Licenciatura Engenharia Informática e Multimédia

Modelação e Programação – MP

Relatório Trabalho Prático 2

Docente Pedro Fazenda

Trabalho realizado por:

Fábio Dias, nº 42921

Índice

1. pack1ColeccoesComHeranca

1.1. Obra

No constructor desta classe, o nome recebido é válido pelo método *validarNome* que verifica cada caractere da *String* recebida. Se este não for uma letra, espaço ou número, o nome é inválido. De seguida, removo os espaços extra com a função *removeExtraSpaces* e o nome é aceite.

Os métodos *getNumPaginas* e *getPreco* são abstractas, logo não possuem corpo.

Implemento o *validarNomes* que, com o auxílio do *validarNome*, valida os nomes recebidos num *array*. Também é implementado o método *haRepeticoes* que recebe um *array* de *String*'s e que verifica se existe repetição de nomes.

Também é desenvolvida a função *toString* que devolve o nome da Obra e, caso este seja Livro, devolve o número de páginas e o seu preço. O método *print* escreve a *String* recebida como argumento, seguido do resultado do *toString*.

Finalmente, é implementada a função *equals* que verifica se o objeto recebido como argumento é uma instância de Obra e se o seu título é igual ao título da Obra atual.

Código:

```
package tp2.pack1ColeccoesComHeranca;

public abstract class Obra {

	private String titulo;

	/**
	 * Constructor
	 */
	public Obra(String titulo) {
		// título (valida título e guarda-o)
		if(!validarNome(titulo))
		{
			throw new IllegalArgumentException("O título é inválido");
		}

		this.titulo = removeExtraSpaces(titulo);
	}

	/**
	 * Devolve o título da obra
	 */
	public String getTitulo() {

		return titulo;
	}

	/**
```

```
     * Devolve o número de páginas da obra
     */
    public abstract int getNumPaginas();

    /**
     * Devolve o preço da obra
     */
    public abstract float getPreco();
```

```
	/**
	 * Deve devolver true se o array conter apenas nomes válidos. Cada nome deve ser
	 * validado pelo método validarNome
	 */
	public static boolean validarNomes(String[] nomes) {

		for(int nomeIndex = 0; nomeIndex < nomes.length; nomeIndex++)
		{
			if(!validarNome(nomes[nomeIndex]))
			{
				return false;
			}
		}

		return true;
	}

	/**
	 * Um nome válido se não for null e conter pelo menos uma letra
	 *  (Character.isLetter) e só conter letras e espaços (Character.isWhitespace)
	 */
	public static boolean validarNome(String nome) {

		if(nome == null || nome.length() == 0)
		{
			return false;
		}

		int letterCounter = 0;

		for(int charIndex = 0; charIndex < nome.length(); charIndex++)
	{
		if(!Character.isLetter(nome.charAt(charIndex)) &&
!Character.isWhitespace(nome.charAt(charIndex)) &&
!Character.isDigit(nome.charAt(charIndex)))
			{
			return false;
			}

		if(Character.isLetter(nome.charAt(charIndex)))
		{
			letterCounter++;
		}
	}

		if(letterCounter == 0)
		{
			return false;
		}

		return true;
```

```
	}

	/**
	 * Recebe um nome já previamente validado, ou seja só com letras ou
espaços.
	 * Deve devolver o mesmo nome mas sem espaços (utilizar trim e
	 * Character.isWhitespace) no início nem no fim e só com um espaço '
' entre
	 * cada nome. Deve utilizar um StringBuilder para ir contendo o nome
já
	 * corrigido
	 */
	public static String removeExtraSpaces(String nome) {

		StringBuilder builder = new StringBuilder();
	nome = nome.trim();

	int charIndex = 0;
	int whitespaceInARow = 0;

	while(charIndex < nome.length())
	{
		if(Character.isWhitespace(nome.charAt(charIndex)))
		{
			whitespaceInARow++;

			if(whitespaceInARow < 2)
			{
				builder.append(nome.charAt(charIndex));
			}
		}
		else
		{
			whitespaceInARow = 0;
			builder.append(nome.charAt(charIndex));
		}

		charIndex++;
	}

	return builder.toString();
	}
```

```
	/**
	 * Método que verifica se há elementos repetidos. O array recebido não contém
	 * nulls.
	 */
	public static boolean haRepeticoes(String[] elems) {

		for(int elementIndex = 0; elementIndex < elems.length; elementIndex++)
		{
			String elementToBeCompared = elems[elementIndex];
			int elementToBeComparedIndex = elementIndex;

			for(int compareElementIndex = 0; compareElementIndex < elems.length; compareElementIndex++)
			{
				if(elementToBeComparedIndex != compareElementIndex)
				{

if(elementToBeCompared.equals(elems[compareElementIndex]))
					{
						return true;
					}
				}
			}
		}

		return false;
	}

	/**
	 * Devolve uma string com a informação da obra (ver outputs desejados e método
	 * toString de Livro)
	 */
	public String toString() {
		//TODO
		String toString = "";

		if(this instanceof Livro)
		{
			toString = titulo + ", " + getNumPaginas() + "p " + getPreco();
		}

		if(this instanceof Coleccao)
		{
			toString = titulo;
		}

		return toString;
	}
```

```
	/**
	 * Deve mostrar na consola a informação da obra (toString) precedida
do prefixo
	 * recebido
	 */
	public void print(String prefix) {
		System.out.println(prefix + toString());
	}

	/**
	 * O Object recebido é igual, se não for null, se for uma obra e se
tiver o
	 * mesmo título que o título da obra corrente
	 */
	public boolean equals(Object l) {

		if(l == null || !(l instanceof Obra))
		{
			return false;
		}

		if(!((Obra) l).getTitulo().equals(getTitulo()))
		{
			return false;
		}

		return true;
	}

}
```

1.2. IObra

Pelo IDE Eclipse, é possível extrair a interface de uma classe automaticamente.

Os únicos métodos considerados para a interface foram o *getPreco* e *getNumPaginas* dado que a sua implementação depende da classe em que se encontram.

Código IObra:

```
package tp2.pack1ColeccoesComHeranca;

public interface IObra {

	/**
	 * Devolve o número de páginas da obra
	 */
	int getNumPaginas();

	/**
	 * Devolve o preço da obra
	 */
	float getPreco();

}
```

Código Obra depois de extraída a interface:

```
package tp2.pack1ColeccoesComHeranca;

public abstract class Obra {

	private String titulo;

	/**
	 * Constructor
	 */
	public Obra(String titulo) {
		// título (valida título e guarda-o)
		if(!validarNome(titulo))
		{
			throw new IllegalArgumentException("O título é inválido");
		}

		this.titulo = removeExtraSpaces(titulo);
	}
```

```java
	/**
	 * Devolve o título da obra
	 */
	public String getTitulo() {

		return titulo;
	}

	/**
	 * Deve devolver true se o array conter apenas nomes válidos. Cada nome deve ser
	 * validado pelo método validarNome
	 */
	public static boolean validarNomes(String[] nomes) {

		for(int nomeIndex = 0; nomeIndex < nomes.length; nomeIndex++)
		{
			if(!validarNome(nomes[nomeIndex]))
			{
				return false;
			}
		}

		return true;
	}
```

```
	/**
	 * Um nome válido se não for null e conter pelo menos uma letra
	   *   (Character.isLetter)   e   só   conter   letras   e   espaços
(Character.isWhitespace)
	 */
	public static boolean validarNome(String nome) {

		if(nome == null || nome.length() == 0)
		{
			return false;
		}

		int letterCounter = 0;

		for(int charIndex = 0; charIndex < nome.length(); charIndex++)
	{
		if(!Character.isLetter(nome.charAt(charIndex))                  &&
!Character.isWhitespace(nome.charAt(charIndex))                         &&
!Character.isDigit(nome.charAt(charIndex)))
			{
			return false;
			}

		if(Character.isLetter(nome.charAt(charIndex)))
		{
			letterCounter++;
		}
	}

		if(letterCounter == 0)
		{
			return false;
		}

		return true;
	}
```

```java
	/**
	 * Recebe um nome já previamente validado, ou seja só com letras ou espaços.
	 * Deve devolver o mesmo nome mas sem espaços (utilizar trim e
	 * Character.isWhitespace) no início nem no fim e só com um espaço ' ' entre
	 * cada nome. Deve utilizar um StringBuilder para ir contendo o nome já
	 * corrigido
	 */
	public static String removeExtraSpaces(String nome) {

		StringBuilder builder = new StringBuilder();
	nome = nome.trim();

	int charIndex = 0;
	int whitespaceInARow = 0;

	while(charIndex < nome.length())
	{
		if(Character.isWhitespace(nome.charAt(charIndex)))
		{
			whitespaceInARow++;

			if(whitespaceInARow < 2)
			{
				builder.append(nome.charAt(charIndex));
			}
		}
		else
		{
			whitespaceInARow = 0;
			builder.append(nome.charAt(charIndex));
		}

		charIndex++;
	}

	return builder.toString();
	}
```

```
	/**
	 * Método que verifica se há elementos repetidos. O array recebido não contém
	 * nulls.
	 */
	public static boolean haRepeticoes(String[] elems) {

		for(int elementIndex = 0; elementIndex < elems.length; elementIndex++)
		{
			String elementToBeCompared = elems[elementIndex];
			int elementToBeComparedIndex = elementIndex;

			for(int compareElementIndex = 0; compareElementIndex < elems.length; compareElementIndex++)
			{
				if(elementToBeComparedIndex != compareElementIndex)
				{

if(elementToBeCompared.equals(elems[compareElementIndex]))
					{
						return true;
					}
				}
			}
		}

		return false;
	}

	/**
	 * Devolve uma string com a informação da obra (ver outputs desejados e método
	 * toString de Livro)
	 */
	public String toString() {

		return titulo;
	}

	/**
	 * Deve mostrar na consola a informação da obra (toString) precedida do prefixo
	 * recebido
	 */
	public void print(String prefix) {
		System.out.println(prefix + toString());
	}
```

```
	/**
	 * O Object recebido é igual, se não for null, se for uma obra e se
tiver o
	 * mesmo título que o título da obra corrente
	 */
	public boolean equals(Object l) {

		if(l == null || !(l instanceof Obra))
		{
			return false;
		}

		if(!((Obra) l).getTitulo().equals(getTitulo()))
		{
			return false;
		}

		return true;
	}

}
```

1.3. Livro

Quando esta classe é declarada, esta é estendida da classe Obra e implementa a interface IObra. No constructor da classe Livro, é passado o título ao constructor da Obra, a classe-pai de Livro. O número de páginas tem de ser positivo, assim como o preço. Os autores não podem possuir *nulls* e têm de conter, pelo menos, uma letra. Só podem conter letras e espaços. São chamadas as funções *removeExtraSpaces*, *validarNomes* e *haRepeticoes* da super classe para validar os autores.

Dado que implementamos a interface IObra, somos obrigados a implementar os métodos *getNumPaginas* e o *getPreco*.

contemAutor verifica se existe o autor passado como argumento no *array* de autores. *getAutores* devolve o *array* de autores.

toString devolve a informação do Livro. *equals* verifica se o objeto recebido como argumento é uma instância da classe Livro e se o seu nome é igual.

Por fim, temos o método *main* onde criamos várias instâncias de livros e são testados todos os métodos.

Código:

```
package tp2.pack1ColeccoesComHeranca;

/**
 * Classe que deverá suportar um livro
 */
public class Livro extends Obra implements IObra{

	// número de páginas
	private int numPaginas;

	// preço do livro
	private float preco;

	// array de autores, este array não deve ter nulls
	private String[] autores;
```

```java
    /**
     * Deve criar um novo livro com os dados recebidos. O número de páginas não
     * pode ser menor que 1. O preço não pode ser negativo. O array de autores
     * não deve conter nem nulls e deve conter pelo menos um autor válido. Não
     * pode haver repetições dos nomes dos autores, considera-se os nomes sem os
     * espaços extra (ver removeExtraSpaces). Este método deve utilizar os
     * métodos auxiliares existentes. Em caso de nome inválido deve lançar uma
     * excepção de IllegalArgumentException com a indicação do erro ocorrido
     */
    public Livro(String titulo, int numPaginas, float preco, String[] autores) {

        super(titulo);

        //Numero de Paginas
        if(numPaginas < 1)
        {
            throw new IllegalArgumentException("O n° de páginas não pode ser negativo");
            //Acho que devia ficar "Tem de ter pelo menos uma página".
        }

        this.numPaginas = numPaginas;

        //Preco
        if(preco < 0)
        {
                throw new IllegalArgumentException("O preço não pode ser negativo");
        }

        this.preco = preco;
```

```
        //Autores
        if(autores == null)
        {
            throw new IllegalArgumentException("O array de autores não pode
ser null");
        }

        String[] novosAutores = new String[autores.length];

                for(int autoresIndex = 0; autoresIndex < autores.length;
autoresIndex++)
        {
            if(autores[autoresIndex] == null)
            {
                  throw new IllegalArgumentException("O array de autores não
pode conter nulls");
            }

            novosAutores[autoresIndex]                                    =
removeExtraSpaces(autores[autoresIndex]);
        }

        if(!validarNomes(novosAutores))
            {
                  throw new IllegalArgumentException("O array de autores só
pode conter nomes válidos");
            }

        if(haRepeticoes(novosAutores))
        {
            throw new IllegalArgumentException("O array de autores contém
autores repetidos");
        }

        this.autores = novosAutores;
      }

      /**
       * Devolve o número de páginas do livro
       */
      public int getNumPaginas() {

            return numPaginas;
      }

      /**
       * Devolve o preço do livro
       */
      public float getPreco() {

            return preco;
      }
```

```
	/**
	 * Devolve true se o autor recebido existe como autor do livro. O
nome
	 * recebido não contém espaços extra.
	 */
	public boolean contemAutor(String autorNome) {

		for(String autor : autores)
		{
			if(autorNome.equals(autor))
			{
				return true;
			}
		}

		return false;
	}

	/**
	 * Devolve uma cópia do array de autores do livro
	 */
	public String[] getAutores() {

		return autores.clone();
	}

	/**
	 * Devolve uma string com a informação do livro (ver outputs
desejados)
	 */
	public String toString() {
		String toString = super.toString() + ", " + getNumPaginas() + "p
" + getPreco() + " [";

		for(int index = 0; index < autores.length; index++)
		{
			toString += autores[index];

			if(index != autores.length - 1)
			{
				toString += ", ";
			}
		}

		toString += "]";

		return toString;
	}
```

```
	/**
	 * Iguais se equais no contexto de obra e se o objecto recebido for
um Livro.
	 * Deve utilizar o método equals de Obra
	 */
	public boolean equals(Object l) {
		return (super.equals(l) && l instanceof Livro);
	}



	/**
	 * main
	 */
	public static void main(String[] args) {

		// constructor e toString
		Livro l = new Livro("Viagem aos Himalaias", 340, 12.3f,
				new String[] { "João Mendonça", "Mário Andrade" });
		System.out.println("Livro -> " + l);
		l.print("");
		l.print("-> ");
		System.out.println();

		// contém autor
		String autorNome = "Mário Andrade";
		System.out.println("Livro com o autor " + autorNome + "? -> "
				+ l.contemAutor(autorNome));
		autorNome = "Mário Zambujal";
		System.out.println("Livro com o autor " + autorNome + "? -> "
				+ l.contemAutor(autorNome));
		System.out.println();

		// equals
		System.out.println("Livro: " + l);
		System.out.println("equals Livro: " + l);
		System.out.println(" -> " + l.equals(l));

		Livro l2 = new Livro("Viagem aos Himalaias", 100, 10.3f,
				new String[] { "Vitor Záspara" });
		System.out.println("Livro: " + l);
		System.out.println("equals Livro: " + l2);
		System.out.println(" -> " + l.equals(l2));
		System.out.println();
```

```
		// testes que dão excepção - mostra-se a excepção

		// livro lx1
		System.out.println("Livro lx1: ");
		try {
			Livro lx1 = new Livro("Viagem aos Himalaias", -1, 12.3f,
					new String[] { "João Mendonça", "Mário
Andrade" });
			System.out.println("Livro lx1: " + lx1);
		} catch (IllegalArgumentException ex) {
			ex.printStackTrace();
		}
		System.out.println();

		// livro lx2
		System.out.println("Livro lx2: ");
		try {
			Livro lx2 = new Livro("Viagem aos Himalaias", 200, -12.3f,
					new String[] { "João Mendonça", "Mário
Andrade" });
			System.out.println("Livro lx2: " + lx2);
		} catch (IllegalArgumentException ex) {
			ex.printStackTrace();
		}
		System.out.println();

		// livro lx3
		System.out.println("Livro lx3: ");
		try {
			Livro lx3 = new Livro(null, 200, -12.3f,
					new String[] { "João Mendonça", "Mário
Andrade" });
			System.out.println("Livro lx3: " + lx3);
		} catch (IllegalArgumentException ex) {
			ex.printStackTrace();
		}
		System.out.println();

		// livro lx4
		System.out.println("Livro lx4: ");
		try {
			Livro lx4 = new Livro("Viagem aos Himalaias", 200, 12.3f,
					new String[] { "João Mendonça", "Mário
Andrade",
							"João Mendonça" });
			System.out.println("Livro lx4: " + lx4);
		} catch (IllegalArgumentException ex) {
			ex.printStackTrace();
		}
```

```
		//Meus Testes:
		System.out.println();
		System.out.println("------------------------");
		System.out.println("Meus Testes:");
		System.out.println("------------------------");

		//Testar Construtor
		System.out.println();
		System.out.println(" -> Testar Construtor");
		System.out.println();

		//Título
		System.out.println("################################## - Título -
##################################");
		System.out.println();

		System.out.println("     #Titulo é null");
		try {
			Livro meuLivroErro = new Livro(null, 100, 20.0f, new
String[] {"Autor Teste"});
		}
		catch(IllegalArgumentException ex)
		{
			ex.printStackTrace();
		}
		System.out.println();

		System.out.println("     #Titulo está vazio");
		try {
			Livro meuLivroErro = new Livro("", 100, 20.0f, new
String[] {"Autor Teste"});
		}
		catch(IllegalArgumentException ex)
		{
			ex.printStackTrace();
		}
		System.out.println();

		System.out.println("     #Titulo apenas contém espaços");
		try {
			Livro meuLivroErro = new Livro("     ", 100, 20.0f, new
String[] {"Autor Teste"});
		}
		catch(IllegalArgumentException ex)
		{
			ex.printStackTrace();
		}
		System.out.println();
```

```
            System.out.println("     #Titulo apenas contém um ponto");
            try {
                  Livro meuLivroErro = new Livro(".", 100, 20.0f, new
String[] {"Autor Teste"});
            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }
            System.out.println();

            System.out.println("     #Titulo possui apenas números");
            try {
                  Livro meuLivroErro = new Livro("1312", 100, 20.0f, new
String[] {"Autor Teste"});
            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }
            System.out.println();

            System.out.println("     #Titulo tem caracteres, espaços extra e
números");
            try {
                  Livro meuLivroCorrecto = new Livro("Teste Correcto Espacos
W    Extra Corta 010101110101010", 100, 20.0f, new String[] {"Autor Teste"});

                  meuLivroCorrecto.print(" ");
            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }
            System.out.println();

            System.out.println("################################        -xXx
Título xXx- ################################");

            System.out.println();
            System.out.println();
```

```
            //Páginas
            System.out.println("################################  -  Páginas
- ################################");
            System.out.println();

            System.out.println("     #Páginas são 0");
            try {
                  Livro meuLivroErro = new Livro("Título Exemplo", 0, 20.0f,
new String[] {"Autor Teste"});
            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }
            System.out.println();

            System.out.println("     #Páginas é negativo");
            try {
                  Livro meuLivroErro = new Livro("Título Exemplo", -100,
20.0f, new String[] {"Autor Teste"});
            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }
            System.out.println();

            System.out.println("     #Páginas é 40");
            try {
                  Livro meuLivroCorrecto = new Livro("Título Exemplo", 40,
20.0f, new String[] {"Autor Teste"});
                  meuLivroCorrecto.print(" ");
            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }
            System.out.println();

            System.out.println("################################       -xXx
Páginas xXx- ################################");

            System.out.println();
            System.out.println();
```

```
            //Preco
            System.out.println("################################  -  Preco  -
################################");
            System.out.println();

            System.out.println("     #Preco é 0");
            try {
                  Livro meuLivroErro = new Livro("Título Exemplo", 40, 0f,
new String[] {"Autor Teste"});
            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }
            System.out.println();

            System.out.println("     #Preco é negativo");
            try {
                  Livro meuLivroErro = new Livro("Título Exemplo", 40,
-10.0f, new String[] {"Autor Teste"});
            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }
            System.out.println();

            System.out.println("     #Preco é 20");
            try {
                  Livro meuLivroCorrecto = new Livro("Título Exemplo", 40,
20f, new String[] {"Autor Teste"});
                  meuLivroCorrecto.print(" ");
            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }
            System.out.println();

            System.out.println("################################ -xXx Preco
xXx- ################################");

            System.out.println();
            System.out.println();
```

```
            //Autores
            System.out.println("################################  -  Autores
- ################################");
            System.out.println();

            System.out.println("     #Autores é null");
            try {
                  Livro meuLivroErro = new Livro("Título Exemplo", 40, 20f,
null);
            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }
            System.out.println();

            System.out.println("     #Autores contém apenas um null");
            try {
                  Livro meuLivroErro = new Livro("Título Exemplo", 40,
20.0f, new String[] {null});
            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }
            System.out.println();

            System.out.println("     #Autores contém um null");
            try {
                  Livro meuLivroCorrecto = new Livro("Título Exemplo", 40,
20f, new String[] {"Autor Teste", null});
                  meuLivroCorrecto.print(" ");
            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }
            System.out.println();

            System.out.println("      #Autores contém uma entrada apenas com
espaços");
            try {
                  Livro meuLivroErro = new Livro("Título Exemplo", 40,
20.0f, new String[] {"Autor Teste", "     ", "Autora Teste"});

            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }
            System.out.println();
```

```
            System.out.println("     #Autores contém nomes repetidos");
            try {
                  Livro meuLivroErro = new Livro("Título Exemplo", 40,
20.0f, new String[] {"Autor Teste", "Autor Teste"});
            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }
            System.out.println();


            System.out.println("      #Autores contém nomes não repetidos e
válidos");
            try {
                  Livro meuLivroCorrecto = new Livro("Título Exemplo", 40,
20f, new String[] {"Autor Teste", "Autora Teste"});
                  meuLivroCorrecto.print(" ");
            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }
            System.out.println();

            System.out.println("################################       -xXx
Autores xXx- ################################");

            System.out.println();
            System.out.println();

            //Testar Métodos
            System.out.println();
            System.out.println(" -> Testar Métodos");
            System.out.println();

            Livro livroTeste1 = new Livro("Teste 1", 50, 15f, new String[]
{"Afonso Risco"});
            Livro livroTeste2 = new Livro("Teste 2", 20, 5.5f, new String[]
{"Beatriz Turim", "Gabriel Caparra"});
            Livro livroTeste3 = new Livro("Teste 3", 99, 18.79f, new
String[] {"Filomena T Real"});
            Livro livroTeste4 = new Livro("Teste 4", 24, 0.99f, new String[]
{"Miguel Luz", "Sara Luz", "Marco Luz"});
            Livro livroTeste5 = new Livro("Teste 5", 111, 21.31f, new
String[] {"Xavier Terno", "Telmo Guilherme", "Lucinda Ferro", "Diogo
Ventura"});
```

```
            //getNumPaginas
            System.out.println("################################## -
getNumPaginas - ##################################");
            System.out.println();

            System.out.println("Expectável  :  50    |  Recebido  :  " +
livroTeste1.getNumPaginas());
            System.out.println("Expectável  :  20    |  Recebido  :  " +
livroTeste2.getNumPaginas());
            System.out.println("Expectável  :  90    |  Recebido  :  " +
livroTeste3.getNumPaginas());
            System.out.println("Expectável  :  24    |  Recebido  :  " +
livroTeste4.getNumPaginas());
            System.out.println("Expectável  :  111   |  Recebido  :  " +
livroTeste5.getNumPaginas());

            System.out.println();
            System.out.println("################################## -xXx
getNumPaginas xXx- ##################################");

            System.out.println();
            System.out.println();

            //getPreco
            System.out.println("################################## - getPreco
- ##################################");
            System.out.println();

            System.out.println("Expectável  :  15.0    |  Recebido  :  " +
livroTeste1.getPreco());
            System.out.println("Expectável  :  5.5     |  Recebido  :  " +
livroTeste2.getPreco());
            System.out.println("Expectável  :  18.79   |  Recebido  :  " +
livroTeste3.getPreco());
            System.out.println("Expectável  :  0.99    |  Recebido  :  " +
livroTeste4.getPreco());
            System.out.println("Expectável  :  21.31   |  Recebido  :  " +
livroTeste5.getPreco());

            System.out.println();
            System.out.println("################################## -xXx
getPreco xXx- ##################################");

            System.out.println();
            System.out.println();
```

```
            //contemAutor
            System.out.println("################################## -
contemAutor - ##################################");
            System.out.println();

            System.out.println("Livro Teste 1 Contem Afonso?
Expectável : false | Recebido : " + livroTeste1.contemAutor("Afonso"));
            System.out.println("Livro Teste 1 Contem Benjamim Fernandes?
Expectável : false | Recebido : " + livroTeste1.contemAutor("Benjamim
Fernades"));
            System.out.println("Livro Teste 1 Contem Afonso Risco?
Expectável : true | Recebido : " + livroTeste1.contemAutor("Afonso
Risco"));
            System.out.println("Livro Teste 2 Contem Beatriz Caparra?
Expectável : false | Recebido : " + livroTeste2.contemAutor("Beatriz
Caparra"));
            System.out.println("Livro Teste 2 Contem Beatriz Tirum?
Expectável : false | Recebido : " + livroTeste2.contemAutor("Beatriz
Tirum"));
            System.out.println("Livro Teste 2 Contem Gabriel Caparra?
Expectável : true | Recebido : " + livroTeste2.contemAutor("Gabriel
Caparra"));
            System.out.println("Livro Teste 3 Contem Filomena A Real?
Expectável : false | Recebido : " + livroTeste3.contemAutor("Filomena A
Real"));
            System.out.println("Livro Teste 3 Contem Filomena T Real?
Expectável : true | Recebido : " + livroTeste3.contemAutor("Filomena T
Real"));
            System.out.println("Livro Teste 4 Contem Rui Luz?
Expectável : false | Recebido : " + livroTeste4.contemAutor("Rui Luz"));
            System.out.println("Livro Teste 4 Contem Sara Luz?
Expectável : true | Recebido : " + livroTeste4.contemAutor("Sara Luz"));
            System.out.println("Livro Teste 5 Contem Xavi?
Expectável : false | Recebido : " + livroTeste5.contemAutor("Xavi"));
            System.out.println("Livro Teste 5 Contem Diogo Ventura?
Expectável : true | Recebido : " + livroTeste5.contemAutor("Diogo
Ventura"));

            System.out.println();
            System.out.println("################################## -xXx
contemAutor xXx- ##################################");

            System.out.println();
            System.out.println();
```

```
            //getAutores
            System.out.println("################################## -
getAutores - ##################################");
            System.out.println();

            String[] autoresLivroTeste1 = livroTeste1.getAutores();
            String[] autoresLivroTeste2 = livroTeste2.getAutores();
            String[] autoresLivroTeste3 = livroTeste3.getAutores();
            String[] autoresLivroTeste4 = livroTeste4.getAutores();
            String[] autoresLivroTeste5 = livroTeste5.getAutores();

            System.out.print("Autores Livro Teste 1 > Expectável : {Afonso
Risco} | Recebido : ");

            for(int autoresIndex = 0; autoresIndex <
autoresLivroTeste1.length; autoresIndex++)
            {
                  if(autoresIndex == 0)
                  {
                        System.out.print("{");
                  }

                  System.out.print(autoresLivroTeste1[autoresIndex]);

                  if(autoresIndex == autoresLivroTeste1.length - 1)
                  {
                        System.out.println("}");
                  }
                  else
                  {
                        System.out.print(", ");
                  }
            }
```

```
            System.out.print("Autores Livro Teste 2 > Expectável : {Beatriz
Turim, Gabriel Caparra} | Recebido : ");

            for(int autoresIndex = 0; autoresIndex <
autoresLivroTeste2.length; autoresIndex++)
            {
                  if(autoresIndex == 0)
                  {
                        System.out.print("{");
                  }

                  System.out.print(autoresLivroTeste2[autoresIndex]);

                  if(autoresIndex == autoresLivroTeste2.length - 1)
                  {
                        System.out.println("}");
                  }
                  else
                  {
                        System.out.print(", ");
                  }
            }

            System.out.print("Autores Livro Teste 3 > Expectável : {Filomena
T Real} | Recebido : ");

            for(int autoresIndex = 0; autoresIndex <
autoresLivroTeste3.length; autoresIndex++)
            {
                  if(autoresIndex == 0)
                  {
                        System.out.print("{");
                  }

                  System.out.print(autoresLivroTeste3[autoresIndex]);

                  if(autoresIndex == autoresLivroTeste3.length - 1)
                  {
                        System.out.println("}");
                  }
                  else
                  {
                        System.out.print(", ");
                  }
            }
```

```
            System.out.print("Autores Livro Teste 4 > Expectável : {Miguel
Luz, Sara Luz, Marco Luz} | Recebido : ");

            for(int autoresIndex = 0; autoresIndex <
autoresLivroTeste4.length; autoresIndex++)
            {
                  if(autoresIndex == 0)
                  {
                        System.out.print("{");
                  }

                  System.out.print(autoresLivroTeste4[autoresIndex]);

                  if(autoresIndex == autoresLivroTeste4.length - 1)
                  {
                        System.out.println("}");
                  }
                  else
                  {
                        System.out.print(", ");
                  }
            }

            System.out.print("Autores Livro Teste 5 > Expectável : {Xavier
Terno, Telmo Guilherme, Lucinda Ferro, Diogo Ventura} | Recebido : ");

            for(int autoresIndex = 0; autoresIndex <
autoresLivroTeste5.length; autoresIndex++)
            {
                  if(autoresIndex == 0)
                  {
                        System.out.print("{");
                  }

                  System.out.print(autoresLivroTeste5[autoresIndex]);

                  if(autoresIndex == autoresLivroTeste5.length - 1)
                  {
                        System.out.println("}");
                  }
                  else
                  {
                        System.out.print(", ");
                  }
            }

            System.out.println();
            System.out.println("################################## -xXx
getAutores xXx- ##################################");

            System.out.println();
            System.out.println();
```

```
            //toString
            System.out.println("################################## - toString
- ##################################");
            System.out.println();

            System.out.println("Livro Teste 1 > Expectável : Teste 1, 50p
15.0 [Afonso Risco] | Recebido : " + livroTeste1.toString());
            System.out.println("Livro Teste 2 > Expectável : Teste 2, 20p
5.5 [Beatriz Turim, Gabriel Caparra] | Recebido : " +
livroTeste2.toString());
            System.out.println("Livro Teste 3 > Expectável : Teste 3, 99p
18.79 [Filomena T Real] | Recebido : " + livroTeste3.toString());
            System.out.println("Livro Teste 4 > Expectável : Teste 4, 24p
0.99 [Miguel Luz, Sara Luz, Marco Luz] | Recebido : " +
livroTeste4.toString());
            System.out.println("Livro Teste 5 > Expectável : Teste 5, 111p
21.31 [Xavier Terno, Telmo Guilherme, Lucinda Ferro, Diogo Ventura] |
Recebido : " + livroTeste5.toString());

            System.out.println();
            System.out.println("################################       -xXx
toString xXx- ################################");

            System.out.println();
            System.out.println();

            //equals
            System.out.println("################################## - equals -
##################################");
            System.out.println();


            System.out.println(" # Teste 1");
            Livro livroEqualsTeste1 = new Livro("Teste 1", 50, 15f, new
String[] {"Afonso Risco"});
            Livro livroEqualsTeste2 = new Livro("Teste 1", 30, 7.55f, new
String[] {"Carlos Noruega"});
            Livro livroEqualsTeste3 = new Livro("Teste 0", 30, 15f, new
String[] {"Afonso Risco"});
            Livro livroEqualsTeste4 = new Livro("Teste 99", 120, 29.98f, new
String[] {"Gustavo Albuquerque", "Margarida Feliz"});
```

```
            System.out.println(livroTeste1.toString() + " Equals " +
livroEqualsTeste1.toString() + " ? Expectável : true | Recebido : " +
livroTeste1.equals(livroEqualsTeste1));
            System.out.println(livroTeste1.toString() + " Equals " +
livroEqualsTeste2.toString() + " ? Expectável : true | Recebido : " +
livroTeste1.equals(livroEqualsTeste2));
            System.out.println(livroTeste1.toString() + " Equals " +
livroEqualsTeste3.toString() + " ? Expectável : false | Recebido : " +
livroTeste1.equals(livroEqualsTeste3));
            System.out.println(livroTeste1.toString() + " Equals " +
livroEqualsTeste4.toString() + " ? Expectável : false | Recebido : " +
livroTeste1.equals(livroEqualsTeste4));
            System.out.println();

            System.out.println(" # Teste 2");

            livroEqualsTeste1 = new Livro("Teste 2", 20, 5.5f, new String[]
{"Beatriz Turim", "Gabriel Caparra"});
            livroEqualsTeste2 = new Livro("Teste 2", 30, 7.55f, new String[]
{"Francisca Guerreiro"});
            livroEqualsTeste3 = new Livro("Teste 0", 30, 15f, new String[]
{"Beatriz Turim", "Gabriel Caparra"});
            livroEqualsTeste4 = new Livro("Teste 99", 120, 29.98f, new
String[] {"Augusto Oliveira", "Sofia Milhar"});

            System.out.println(livroTeste2.toString() + " Equals " +
livroEqualsTeste1.toString() + " ? Expectável : true | Recebido : " +
livroTeste2.equals(livroEqualsTeste1));
            System.out.println(livroTeste2.toString() + " Equals " +
livroEqualsTeste2.toString() + " ? Expectável : true | Recebido : " +
livroTeste2.equals(livroEqualsTeste2));
            System.out.println(livroTeste2.toString() + " Equals " +
livroEqualsTeste3.toString() + " ? Expectável : false | Recebido : " +
livroTeste2.equals(livroEqualsTeste3));
            System.out.println(livroTeste2.toString() + " Equals " +
livroEqualsTeste4.toString() + " ? Expectável : false | Recebido : " +
livroTeste2.equals(livroEqualsTeste4));
            System.out.println();

            System.out.println(" # Teste 3");

            livroEqualsTeste1 = new Livro("Teste 3", 99, 18.79f, new
String[] {"Filomena T Real"});
            livroEqualsTeste2 = new Livro("Teste 3", 30, 7.55f, new String[]
{"Vladimir Rute"});
            livroEqualsTeste3 = new Livro("Teste 0", 30, 15f, new String[]
{"Filipe Sampaio"});
            livroEqualsTeste4 = new Livro("Teste 99", 120, 29.98f, new
String[] {"Frederico Tavares", "David Reboio"});
```

```
            System.out.println(livroTeste3.toString() + " Equals " +
livroEqualsTeste1.toString() + " ? Expectável : true | Recebido : " +
livroTeste3.equals(livroEqualsTeste1));
            System.out.println(livroTeste3.toString() + " Equals " +
livroEqualsTeste2.toString() + " ? Expectável : true | Recebido : " +
livroTeste3.equals(livroEqualsTeste2));
            System.out.println(livroTeste3.toString() + " Equals " +
livroEqualsTeste3.toString() + " ? Expectável : false | Recebido : " +
livroTeste3.equals(livroEqualsTeste3));
            System.out.println(livroTeste3.toString() + " Equals " +
livroEqualsTeste4.toString() + " ? Expectável : false | Recebido : " +
livroTeste3.equals(livroEqualsTeste4));
            System.out.println();

            System.out.println(" # Teste 4");

            livroEqualsTeste1 = new Livro("Teste 4", 24, 0.99f, new String[]
{"Miguel Luz", "Sara Luz", "Marco Luz"});
            livroEqualsTeste2 = new Livro("Teste 4", 30, 7.55f, new String[]
{"Ricardo Silva", "Gilbero Costa"});
            livroEqualsTeste3 = new Livro("Teste 0", 30, 15f, new String[]
{"André Flores"});
            livroEqualsTeste4 = new Livro("Teste 99", 120, 29.98f, new
String[] {"Susana Dias", "Carlos Dias"});

            System.out.println(livroTeste4.toString() + " Equals " +
livroEqualsTeste1.toString() + " ? Expectável : true | Recebido : " +
livroTeste4.equals(livroEqualsTeste1));
            System.out.println(livroTeste4.toString() + " Equals " +
livroEqualsTeste2.toString() + " ? Expectável : true | Recebido : " +
livroTeste4.equals(livroEqualsTeste2));
            System.out.println(livroTeste4.toString() + " Equals " +
livroEqualsTeste3.toString() + " ? Expectável : false | Recebido : " +
livroTeste4.equals(livroEqualsTeste3));
            System.out.println(livroTeste4.toString() + " Equals " +
livroEqualsTeste4.toString() + " ? Expectável : false | Recebido : " +
livroTeste4.equals(livroEqualsTeste4));
            System.out.println();
```

```
            System.out.println(" # Teste 5");

            livroEqualsTeste1 = new Livro("Teste 5", 111, 21.31f, new
String[] {"Xavier Terno", "Telmo Guilherme", "Lucinda Ferro", "Diogo
Ventura"});
            livroEqualsTeste2 = new Livro("Teste 5", 30, 7.55f, new String[]
{"Afonso Dinis", "Luísa Reis"});
            livroEqualsTeste3 = new Livro("Teste 0", 30, 15f, new String[]
{"Paulo Páscoa", "Paulo Trindade"});
            livroEqualsTeste4 = new Livro("Teste 99", 120, 29.98f, new
String[] {"Telma Dias", "Ivo Gerónimo"});

            System.out.println(livroTeste5.toString() + " Equals " +
livroEqualsTeste1.toString() + " ? Expectável : true | Recebido : " +
livroTeste5.equals(livroEqualsTeste1));
            System.out.println(livroTeste5.toString() + " Equals " +
livroEqualsTeste2.toString() + " ? Expectável : true | Recebido : " +
livroTeste5.equals(livroEqualsTeste2));
            System.out.println(livroTeste5.toString() + " Equals " +
livroEqualsTeste3.toString() + " ? Expectável : false | Recebido : " +
livroTeste5.equals(livroEqualsTeste3));
            System.out.println(livroTeste5.toString() + " Equals " +
livroEqualsTeste4.toString() + " ? Expectável : false | Recebido : " +
livroTeste5.equals(livroEqualsTeste4));
            System.out.println();

            System.out.println();
            System.out.println("################################       -xXx
equals xXx- ################################");
      }
}
```

1.4. Coleccao

Na classe Coleccao, o seu constructor recebe uma *String* nome que é validada na classe-pai, Obra. Também recebe um *array* de *String*'s que são os editores e estes só podem ter letras, espaços e números, não podem ser vazios ou *nulls*, o *array* não pode ser *null* e não pode conter nomes repetidos.

Dado que implementamos a interface IObra, somos obrigados a implementar os métodos *getNumPaginas* e o *getPreco*.

addObra verifica se é possível adicionar a Obra passada como argumento, ou seja, se esta não é *null* e se o *array* de Obras ainda possui espaço, assim como se não existe nenhuma Obra com o mesmo título que a recebida.

getIndexOfObra devolve o índice da obra com o título passado como argumento. Caso não encontre, este devolve -1. *remObra* remove a Obra com o mesmo título passado como argumento, caso esta exista.

remAllObra faz o mesmo que o *remObra* mas, a todos os níveis da Coleccao.

getNumObrasFromPerson devolve o número de Obras em que o autor ou editor passsado como argumento, participou. O método *getLivrosComoAutor* devolve um *array* de Livros cujo o autor passado como argumento, participou no seu desenvolvimento. *getAutoresEditores* devolve todos os autores e editores presentes naquela Coleccao.

A função *mergeWithoutRepetition* tem duas variações: uma para *String*'s e outra para Livros. Esta devolve um único *array*, dependendo da variação usada, com todas as *String*'s, de ambos os *array*'s que recebe, num só mas sem repetições. O mesmo para os Livros.

getNumLivros devolve o número de Livros de uma Coleccao. *getProfundidade* devolve o nível mais baixo daquela Coleccao.

O método *equals* verifica se o objeto recebido é uma Coleccao, se o título é idêntico e se os editores são os mesmos.

toString devolve uma *String* com o *toString* da classe-pai, seguido do número de páginas, preço e editores, assim como o número de livros, o número de colecções e a profundidade máxima. *print* imprime o prefixo passado como argumento, seguido de todas as informações daquela colecção.

Por fim, temos o método *main* onde são testadas todas as funções da classe Coleccao.

Código:

```
package tp2.pack1ColeccoesComHeranca;

import java.util.Arrays;
import java.util.Iterator;
```

```java
/**
 * Classe Coleccao, deve conter a descrição de uma colecção, com título, os seus
 * livros, colecções e editores. Deve utilizar herança para guardar os livros e
 * as colecções num só array
 */
public class Coleccao extends Obra implements IObra {
	// prefixo a colocar no início de cada print mais interno que o corrente
	public static final String GENERALPREFIX = "  ";

	// número máximo de obras de uma colecção
	private static int MAXOBRAS = 20;

	// Array de obras, de Livros ou Coleccção, em que estas devem encontrar-se
	// sempre nos menores índices e pela ordem de registo
	private Obra[] obras = new Obra[MAXOBRAS];

	// deverá conter sempre o número de obras na colecção
	private int numObras = 0;

	// Editores, tem as mesmas condicionantes que array de autores na classe
	// livro
	private String[] editores;

	/**
	 * Construtor; o título deve ser guardado e validado na clase obra; o array de
	 * editores devem ser pelo menos um e tem as mesmas restrições que os autores
	 * dos livros;
	 */
	public Coleccao(String titulo, String[] editores) {

		super(titulo);

		//Editores
		if(editores == null)
		{
			throw new IllegalArgumentException("O array de editores é null");
		}
```

```
        String[] novosEditores = new String[editores.length];

              for(int editoresIndex = 0; editoresIndex < editores.length;
editoresIndex++)
        {
            if(editores[editoresIndex] == null)
            {
                  throw new IllegalArgumentException("O array de editores
não pode conter nulls");
            }

            novosEditores[editoresIndex]                                    =
removeExtraSpaces(editores[editoresIndex]);
        }

        if(!validarNomes(novosEditores))
            {
                  throw new IllegalArgumentException("O array de editores só
pode conter nomes válidos");
            }

        if(haRepeticoes(novosEditores))
        {
            throw new IllegalArgumentException("O array de editores contém
editores repetidos");
        }

        this.editores = novosEditores;
      }
```

```
	/**
	 * Obtem o número total de páginas da colecção, páginas dos livros e
das
	 * colecções
	 */
	public int getNumPaginas() {
		int paginas = 0;

		for(int obrasIndex = 0; obrasIndex < numObras; obrasIndex++)
		{
			if(obras[obrasIndex] instanceof Livro)
			{
				paginas += ((Livro)
obras[obrasIndex]).getNumPaginas();
			}

			if(obras[obrasIndex] instanceof Coleccao)
			{
				paginas += ((Coleccao)
obras[obrasIndex]).getNumPaginas();
			}
		}

		return paginas;
	}
```

```
	/**
	 * As colecções com mais de 5000 páginas nos seus livros directos têm um
	 * desconto de 20% nesses livros. As colecções em que o somatório de páginas das
	 * suas subcolecções directas seja igual ou superior ao quádruplo do nº de
	 * páginas da sua subcolecção directa com mais páginas deverão aplicar um
	 * desconto de 10% sobre os preços das suas subcolecções
	 */
	public float getPreco() {

		float precoActual = 0;
		float precoColeccoes = 0;
		int numPaginas = 0;
		int maiorNumPaginas = 0;

		for(int index = 0; index < numObras; index++)
		{
			if(obras[index] instanceof Livro)
			{
				Livro obra = (Livro) obras[index];
				numPaginas += obra.getNumPaginas();
				precoActual += obra.getPreco();
			}
		}

		if(numPaginas > 5000)
		{
			precoActual = precoActual * 0.8f;
		}

		numPaginas = 0;

		for(int index = 0; index < numObras; index++)
		{
			if(obras[index] instanceof Coleccao)
			{
				Coleccao obra = (Coleccao) obras[index];

				if(obra.getNumPaginas() > maiorNumPaginas)
				{
					maiorNumPaginas = obra.getNumPaginas();
				}

				numPaginas += obra.getNumPaginas();
				precoColeccoes += obra.getPreco();
			}
		}
```

```
            if(numPaginas >= 4 * maiorNumPaginas)
            {
                  precoColeccoes = precoColeccoes * 0.9f;
            }

            return precoActual + precoColeccoes;
      }

      /**
       * Adiciona uma obra à colecção se puder, se esta não for null e a
colecção não
       * ficar com obras com iguais no seu nível imediato. Deve utilizar o
método
       * getIndexOfLivro e getIndexOfColeccao
       */
      public boolean addObra(Obra obra) {
            if(numObras == obras.length || obra == null)
            {
                  return false;
            }

            if(getIndexOfObra(obra.getTitulo()) != -1)
            {
                  return false;
            }

            obras[numObras] = obra;
            numObras++;

            return true;
      }

      /**
       * Devolve o index no array de obras onde estiver a obra com o nome
pretendido.
       * Devolve -1 caso não o encontre
       */
      private int getIndexOfObra(String titulo) {

            for(int index = 0; index < numObras; index++)
            {
                  if(obras[index].getTitulo().equals(titulo))
                  {
                        return index;
                  }
            }

            return -1;
      }
```

```
	/**
	 * Remove do array a obra com o título igual ao título recebido. Devolve a obra
	 * removida ou null caso não tenha encontrado a obra. Deve-se utilizar o método
	 * getIndexOfLivro. Recorda-se que as obras ocupam sempre os menores índices, ou
	 * seja, não pode haver nulls entre elas.
	 */
	public Obra remObra(String titulo) {

		int obraARemoverIndex = getIndexOfObra(titulo);

		if(obraARemoverIndex == -1)
		{
			return null;
		}

		Obra obraRemovida = obras[obraARemoverIndex];

		Obra[] obrasSemObraRemovida = new Obra[MAXOBRAS];

		for(int index = 0; index < obraARemoverIndex; index++)
		{
			obrasSemObraRemovida[index] = obras[index];
		}

		for(int index = obraARemoverIndex + 1; index < numObras; index++)
		{
			obrasSemObraRemovida[index - 1] = obras[index];
		}

		obras = obrasSemObraRemovida;
		numObras--;

		return obraRemovida;
	}
```

```
	/**
	 * Remove todas as obras (livros ou colecções) dentro da obra corrente, que
	 * tenham um título igual ou título recebido. Devolve true se removeu pelo menos
	 * uma obra, ou false caso não tenha trealizado qualquer remoção. Deve utilizar
	 * os métodos remObra e remAllObra.
	 */
	public boolean remAllObra(String titulo) {

		boolean removeu = false;
		boolean removeuRecursivo = false;

		if(remObra(titulo) != null)
		{
			removeu = true;
		}

		for(int index = 0; index < numObras; index++)
		{
			if(obras[index] instanceof Coleccao)
			{
				removeuRecursivo = ((Coleccao) obras[index]).remAllObra(titulo);
			}
		}

		if(removeu || removeuRecursivo)
		{
			return true;
		}

		return false;
	}
```

```
	/**
	 * Devolve o nº de obras de uma pessoa. Cada colecção deve contabilizar-se como
	 * uma obra para os editores.
	 */
	public int getNumObrasFromPerson(String autorEditor) {

		int numObrasFromPerson = 0;

		for(int index = 0; index < numObras; index++)
		{
			if(obras[index] instanceof Livro)
			{
				if(((Livro) obras[index]).contemAutor(autorEditor))
				{
					numObrasFromPerson++;
				}
			}
			else if(obras[index] instanceof Coleccao)
			{
				numObrasFromPerson += ((Coleccao) obras[index]).getNumObrasFromPerson(autorEditor);
			}
		}

		for(int editoresIndex = 0; editoresIndex < editores.length; editoresIndex++)
		{
			if(editores[editoresIndex].equals(autorEditor))
			{
				numObrasFromPerson++;
				break;
			}
		}

		return numObrasFromPerson;
	}
```

```
	/**
	 * Deve devolver um novo array, sem repetições, com os livros de que o autor
	 * recebido é autor. O array devolvido não deve conter repetições, para excluir
	 * as repetições devem utilizar o método mergeWithoutRepetitions
	 */
	public Livro[] getLivrosComoAutor(String autorNome) {

		Livro[] livrosDoAutor = new Livro[0];

		for(int index = 0; index < numObras; index++)
		{
			if(obras[index] instanceof Livro)
			{
				if(((Livro) obras[index]).contemAutor(autorNome))
				{
					livrosDoAutor = mergeWithoutRepetitions(livrosDoAutor, new Livro[] {(Livro) obras[index]});
				}
			}
			else if(obras[index] instanceof Coleccao)
			{
				livrosDoAutor = mergeWithoutRepetitions(livrosDoAutor, ((Coleccao) obras[index]).getLivrosComoAutor(autorNome));
			}
		}

		return livrosDoAutor;
	}
```

```
	/**
	 * Deve devolver um array, sem nulls, com todos os autores e editores existentes
	 * na colecção. O resultado não deve conter repetições. Deve utilizar o método
	 * mergeWithoutRepetitions
	 */
	public String[] getAutoresEditores() {

		String[] editoresAutores = editores.clone();

		for(int index = 0; index < numObras; index++)
		{
			if(obras[index] instanceof Livro)
			{
				editoresAutores = mergeWithoutRepetitions(editoresAutores, ((Livro) obras[index]).getAutores());
			}
			else if(obras[index] instanceof Coleccao)
			{
				editoresAutores = mergeWithoutRepetitions(editoresAutores, ((Coleccao) obras[index]).getAutoresEditores());
			}
		}

		return editoresAutores;
	}
```

```
	/**
	 * Método que recebendo dois arrays sem repetições devolve um novo array com
	 * todos os elementos dos arrays recebidos mas sem repetições
	 */
	private static String[] mergeWithoutRepetitions(String[] a1, String[] a2) {

		String[] arraySemRepeticoes = new String[a1.length + a2.length];
		int indexArraySemRepeticoes = 0;

		for(int index = 0; index < a1.length; index++)
		{
			arraySemRepeticoes[indexArraySemRepeticoes] = a1[index];
			indexArraySemRepeticoes++;
		}

		for(String texto : a2)
		{
			boolean repetido = false;

			for(int indexNovoArray = 0; indexNovoArray < indexArraySemRepeticoes; indexNovoArray++)
			{
				if(texto.equals(arraySemRepeticoes[indexNovoArray]))
				{
					repetido = true;
				}
			}

			if(!repetido)
			{
				arraySemRepeticoes[indexArraySemRepeticoes] = texto;
				indexArraySemRepeticoes++;
			}
		}

		String[] arraySemRepeticoesSemNulls = new String[indexArraySemRepeticoes];

		for(int index = 0; index < indexArraySemRepeticoes; index++)
		{
			arraySemRepeticoesSemNulls[index] = arraySemRepeticoes[index];
		}

		return arraySemRepeticoesSemNulls;
	}
```

```
	/**
	 * Método idêntico ao método anterior mas agora com arrays de livros
	 */
	private static Livro[] mergeWithoutRepetitions(Livro[] a1, Livro[] a2)
{

		Livro[] arraySemRepeticoes = new Livro[a1.length + a2.length];
		int indexArraySemRepeticoes = 0;

		for(int index = 0; index < a1.length; index++)
		{
			arraySemRepeticoes[indexArraySemRepeticoes] = a1[index];
			indexArraySemRepeticoes++;
		}

		for(Livro livro : a2)
		{
			boolean repetido = false;

			for(int indexNovoArray = 0; indexNovoArray <
indexArraySemRepeticoes; indexNovoArray++)
			{
				if(livro.equals(arraySemRepeticoes[indexNovoArray]))
				{
					repetido = true;
				}
			}

			if(!repetido)
			{
				arraySemRepeticoes[indexArraySemRepeticoes] = livro;
				indexArraySemRepeticoes++;
			}
		}

		Livro[] arraySemRepeticoesSemNulls = new
Livro[indexArraySemRepeticoes];

		for(int index = 0; index < indexArraySemRepeticoes; index++)
		{
			arraySemRepeticoesSemNulls[index] =
arraySemRepeticoes[index];
		}

		return arraySemRepeticoesSemNulls;
	}
```

```java
/**
 * Devolve o nº de livros dentro da colecção
 */
public int getNumLivros() {
	int numLivros = 0;

	for(Obra obra : obras)
	{
		if(obra instanceof Livro)
		{
			numLivros++;
		}
		else if(obra instanceof Coleccao)
		{
			numLivros += ((Coleccao) obra).getNumLivros();
		}
	}

	return numLivros;
}

/**
 * Devolve o nº de colecções dentro da colecção
 */
public int getNumColeccoes() {
	int numColeccoes = 0;

	for(Obra obra : obras)
	{
		if(obra instanceof Coleccao)
		{
			numColeccoes += ((Coleccao) obra).getNumColeccoes();
			numColeccoes++;
		}
	}

	return numColeccoes;
}
```

```
	/**
	 * Devolve a profundidada de máxima de uma colecção em termos de
colecções
	 * dentro de coleccções: uma colecção c1 com uma colecção c2 dentro,
c1 deve
	 * devolver 2 e c2 deve devolver 1, independentemente do número do
conteúdo de
	 * cada uma.
	 */
	public int getProfundidade() {

		int profundidade = 1;

		for(int obrasIndex = 0; obrasIndex < numObras; obrasIndex++)
		{
			if(obras[obrasIndex] instanceof Coleccao)
			{
				int aux = ((Coleccao)
obras[obrasIndex]).getProfundidade();

				if(aux + 1 > profundidade)
				{
					profundidade = aux + 1;
				}
			}
		}

		return profundidade;
	}

	/**
	 * Duas colecções são iguais se tiverem o mesmo título e a mesma
lista de
	 * editores. Deve utilizar o equals da classe Obra. Para verificar
verificar se
	 * os editores são os mesmos devem utilizar o método
mergeWithoutRepetitions
	 */
	public boolean equals(Object c) {
		return (super.equals(c) && c instanceof Coleccao &&
mergeWithoutRepetitions(editores, ((Coleccao) c).editores).length ==
editores.length);
	}
```

```
	/**
	 * Deve devolver uma string compatível com os outputs desejados
	 */
	public String toString() {

		String toString = super.toString();

		toString += ", " + getNumPaginas() + "p, " + getPreco() + ",
editores [";

		for(int index = 0; index < editores.length; index++)
		{
			toString += editores[index];

			if(index != editores.length - 1)
			{
				toString += ", ";
			}
		}

		toString += "], com " + getNumLivros() + " livros, com " +
getNumColeccoes() + " colecções e com profundidade máxima de " +
getProfundidade();

		return toString;
	}

	/**
	 * Mostra uma colecção segundo os outputs desejados. Deve utilizar o
método
	 * print da classe Obra.
	 */
	public void print(String prefix) {

		System.out.println(prefix + toString());

		for(int index = 0; index < numObras; index++)
		{
			if(obras[index] instanceof Livro)
			{
				obras[index].print(prefix + " ");

			}
			else if(obras[index] instanceof Coleccao)
			{
				((Coleccao) obras[index]).print(prefix + " ");
			}
		}
	}
```

```
	/**
	 * main
	 */
	public static void main(String[] args) {
		Livro l1 = new Livro("Viagem aos Himalaias", 340, 12.3f, new String[] { "João Mendonça", "Mário Andrade" });
		Livro l2 = new Livro("Viagem aos Pirinéus", 270, 11.5f, new String[] { "João Mendonça", "Júlio Pomar" });

		Coleccao c1 = new Coleccao("Primavera", new String[] { "João Mendonça", "Manuel Alfazema" });

		boolean res;

		res = c1.addObra(l1);
		res = c1.addObra(l2);
		System.out.println("c1 -> " + c1);
		c1.print("");
		System.out.println();

		// adicionar um livro com nome de outro já existente
		res = c1.addObra(l2);
		System.out.println("adição novamente de Viagem aos Pirinéus a c1 -> " + res);
		System.out.println("c1 -> " + c1);
		System.out.println();

		// Outra colecção
		Livro l21 = new Livro("Viagem aos Himalaias 2", 340, 12.3f, new String[] { "João Mendonça", "Mário Andrade" });
		Livro l22 = new Livro("Viagem aos Pirinéus 2", 270, 11.5f, new String[] { "João Mendonça", "Júlio Pomar" });

		Coleccao cx2 = new Coleccao("Outono", new String[] { "João Mendonça", "Manuel Antunes" });
		cx2.addObra(l21);
		cx2.addObra(l22);
		System.out.println("cx2 -> " + cx2);
		cx2.print("");
		System.out.println();

		// adicioná-la a c1
		c1.addObra(cx2);
		System.out.println("c1 após adição da colecção cx2 -> " + c1);
		c1.print("");
		System.out.println();

		// get editores autores
		String[] ae = c1.getAutoresEditores();
		System.out.println("Autores editores of c1 -> " + Arrays.toString(ae));
		System.out.println();
```

```
            // getNumObrasFromPerson
            String nome = "João Mendonça";
            int n = c1.getNumObrasFromPerson(nome);
            System.out.println("N° de obras de " + nome + " -> " + n);
            System.out.println();

            // getLivrosComoAutor
            nome = "João Mendonça";
            Livro[] livros = c1.getLivrosComoAutor(nome);
            System.out.println("Livros de " + nome + " -> " +
Arrays.toString(livros));
            System.out.println();
            System.out.println();

            // testes aos métodos getNumLivros, getNumColeccoes e
getProfundidade
            c1.print("");
            System.out.println("N° de livros na colecção " + c1.getTitulo()
+ " -> " + c1.getNumLivros());

            System.out.println("N° de colecções dentro da colecção " +
c1.getTitulo() + " -> " + c1.getNumColeccoes());

            System.out.println("Profundidade da colecção " + c1.getTitulo()
+ " -> " + c1.getProfundidade());
            System.out.println("Profundidade da colecção " + cx2.getTitulo()
+ " -> " + cx2.getProfundidade());
            System.out.println();

            // rem livro
            String nomeLivro = "Viagem aos Himalaias";
            Obra l = c1.remObra(nomeLivro);
            System.out.println("Remoção de " + nomeLivro + " -> " + l);
            c1.print("");

            //Meus Testes:
            System.out.println();
            System.out.println("------------------------");
            System.out.println("Meus Testes:");
            System.out.println("------------------------");

            //Testar Construtor
            System.out.println();
            System.out.println(" -> Testar Construtor");
            System.out.println();
```

```
            //Editores
            System.out.println("################################## - Editores
- ##################################");
            System.out.println();

            System.out.println("     #O array de editores é null");
            try {
                  Coleccao   minhaColeccaoErro   =   new   Coleccao("Coleccao
Teste", null);
            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }
            System.out.println();

            System.out.println("      #O array de editores contem apenas um
nome vazio");
            try {
                  Coleccao   minhaColeccaoErro   =   new   Coleccao("Coleccao
Teste", new String[] {""});
            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }

            System.out.println("     #O array de editores contem um nome com
apenas espaços");
            try {
                  Coleccao   minhaColeccaoErro   =   new   Coleccao("Coleccao
Teste", new String[] {"Editor Teste", "    "});
            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }
            System.out.println();

            System.out.println("        #O array de editores contem nomes
repetidos");
            try {
                  Coleccao   minhaColeccaoErro   =   new   Coleccao("Coleccao
Teste", new String[] {"Editor Teste, Editor Teste"});
            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }
            System.out.println();
```

```
            System.out.println("       #O array de editores nomes válidos e
não repetidos, com espaços extra");
            try {
                  Coleccao    minhaColeccaoCorrecta    =    new    Coleccao("
Teste Correcto  123 Espacos   11   Extra  Corta    123123   ", new String[]
{"   Editor       Teste   ", "Editora   Teste"});
                  minhaColeccaoCorrecta.print("");
            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }
            System.out.println();

            System.out.println("################################       -xXx
Editores xXx- ################################");

            System.out.println();
            System.out.println();

            //Testar Métodos
            System.out.println();
            System.out.println(" -> Testar Métodos");
            System.out.println();

            Coleccao coleccaoPrincipal = new Coleccao("Coleccao Principal",
new String[] {"Bruno Aleixo"});
            Coleccao  coleccaoNivel2_1  =  new  Coleccao("Coleccao  Nivel  2
Numero 1", new String[] {"Telmo Afonso"});
            Coleccao  coleccaoNivel2_2  =  new  Coleccao("Coleccao  Nivel  2
Numero 2", new String[] {"Carlos Fonseca", "José Ribeiro"});
            Coleccao  coleccaoNivel3_1  =  new  Coleccao("Coleccao  Nivel  3
Numero 1", new String[] {"Samuel Torpedo", "Manuel Teixeira"});
            Coleccao  coleccaoNivel3_2  =  new  Coleccao("Coleccao  Nivel  3
Numero 2", new String[] {"Afonso Risco"});

            Livro  livroTeste1  =  new  Livro("Livro  Teste  1",  50,  15f,  new
String[] {"Afonso Risco"});
            Livro  livroTeste2  =  new  Livro("Livro  Teste  2",  20,  5.5f,  new
String[] {"Beatriz Turim", "Gabriel Caparra"});
            Livro  livroTeste3  =  new  Livro("Livro  Teste  3",  99,  18.79f,  new
String[] {"Filomena T Real"});
            Livro  livroTeste4  =  new  Livro("Livro  Teste  4",  24,  0.99f,  new
String[] {"Miguel Luz", "Sara Luz", "Marco Luz"});
            Livro  livroTeste5  =  new  Livro("Livro  Teste  5",  111,  21.31f,  new
String[]  {"Xavier  Terno",  "Telmo  Guilherme",  "Lucinda  Ferro",  "Diogo
Ventura"});
            Livro  livroTeste6  =  new  Livro("Livro  Teste  6",  15,  2.99f,  new
String[] {"Vera Cruz", "Afonso Risco"});
```

```
            //addObra
            System.out.println("################################## - addObra
- ##################################");
            System.out.println();

            System.out.println("   # Adicionar Livro Teste 1, duas vezes a
Coleccao Principal.");
            System.out.println("Adicionar Livro Teste 1 a Coleccao Principal
> Expectável : true | Recebido : " + coleccaoPrincipal.addObra(livroTeste1));
            System.out.println("Adicionar Livro Teste 1 a Coleccao Principal
>      Expectável      :      false      |      Recebido      :      "      +
coleccaoPrincipal.addObra(livroTeste1));
            System.out.println();

            System.out.println("   # Adicionar Coleccao Nivel 2 Numero 1,
duas vezes a Coleccao Principal.");
            System.out.println("Adicionar Coleccao Nivel 2 Numero 1 a
Coleccao Principal > Expectável : true | Recebido : " +
coleccaoPrincipal.addObra(coleccaoNivel2_1));
            System.out.println("Adicionar Coleccao Nivel 2 Numero 1 a
Coleccao Principal > Expectável : false | Recebido : " +
coleccaoPrincipal.addObra(coleccaoNivel2_1));
            System.out.println();

            System.out.println("  # Adicionar Livro Teste 1 e Livro Teste 2,
duas vezes a Coleccao Nivel 2 Numero 1.");
            System.out.println("Adicionar Livro Teste 2 a Coleccao Nivel 2
Numero 1 > Expectável : true | Recebido : " +
coleccaoNivel2_1.addObra(livroTeste2));
            System.out.println("Adicionar Livro Teste 2 a Coleccao Nivel 2
Numero 1 > Expectável : false | Recebido : " +
coleccaoNivel2_1.addObra(livroTeste2));
            System.out.println("Adicionar Livro Teste 3 a Coleccao Nivel 2
Numero 1 > Expectável : true | Recebido : " +
coleccaoNivel2_1.addObra(livroTeste3));
            System.out.println("Adicionar Livro Teste 3 a Coleccao Nivel 2
Numero 1 > Expectável : false | Recebido : " +
coleccaoNivel2_1.addObra(livroTeste3));
            System.out.println();

            System.out.println("   # Adicionar Coleccao Nivel 3 Numero 1,
duas vezes a Coleccao Nivel 2 Numero 2.");
            System.out.println("Adicionar Coleccao Nivel 3 Numero 1 a
Coleccao Nivel 2 Numero 2 > Expectável : true | Recebido : " +
coleccaoNivel2_2.addObra(coleccaoNivel3_1));
            System.out.println("Adicionar Coleccao Nivel 3 Numero 1 a
Coleccao Nivel 2 Numero 2 > Expectável : false | Recebido : " +
coleccaoNivel2_2.addObra(coleccaoNivel3_1));
            System.out.println();
```

```
            System.out.println("   # Adicionar Livro Teste 1, duas vezes a
Coleccao Nivel 3 Numero 1.");
            System.out.println("Adicionar Coleccao Livro Teste 1 a Coleccao
Nivel 3 Numero 1 > Expectável : true | Recebido : " +
coleccaoNivel3_1.addObra(livroTeste1));
            System.out.println("Adicionar Coleccao Livro Teste 1 a Coleccao
Nivel 3 Numero 1 > Expectável : false | Recebido : " +
coleccaoNivel3_1.addObra(livroTeste1));
            System.out.println();

            System.out.println("   # Adicionar Coleccao Nivel 2 Numero 2,
duas vezes a Coleccao Principal.");
            System.out.println("Adicionar Coleccao Nivel 2 Numero 2 a
Coleccao Principal > Expectável : true | Recebido : " +
coleccaoPrincipal.addObra(coleccaoNivel2_2));
            System.out.println("Adicionar Coleccao Nivel 2 Numero 2 a
Coleccao Principal > Expectável : false | Recebido : " +
coleccaoPrincipal.addObra(coleccaoNivel2_2));
            System.out.println();

            System.out.println("  # Adicionar Livro Teste 4 e Livro Teste 5,
duas vezes a Coleccao Principal.");
            System.out.println("Adicionar Livro Teste 4 a Coleccao Principal
> Expectável : true | Recebido : " + coleccaoPrincipal.addObra(livroTeste4));
            System.out.println("Adicionar Livro Teste 4 a Coleccao Principal
>      Expectável      :      false      |      Recebido      :      "      +
coleccaoPrincipal.addObra(livroTeste4));
            System.out.println("Adicionar Livro Teste 5 a Coleccao Principal
> Expectável : true | Recebido : " + coleccaoPrincipal.addObra(livroTeste5));
            System.out.println("Adicionar Livro Teste 5 a Coleccao Principal
>      Expectável      :      false      |      Recebido      :      "      +
coleccaoPrincipal.addObra(livroTeste5));
            System.out.println();

            System.out.println("   # Adicionar Coleccao Nivel 3 Numero 2,
duas vezes a Coleccao Principal.");
            System.out.println("Adicionar Coleccao Nivel 3 Numero 2 a
Coleccao Principal > Expectável : true | Recebido : " +
coleccaoPrincipal.addObra(coleccaoNivel3_2));
            System.out.println("Adicionar Coleccao Nivel 3 Numero 2 a
Coleccao Principal > Expectável : false | Recebido : " +
coleccaoPrincipal.addObra(coleccaoNivel3_2));
            System.out.println();
```

```
            System.out.println("   # Adicionar Livro Teste 6, duas vezes a
Nivel 3 Numero 2.");
            System.out.println("Adicionar Livro Teste 6 a Nivel 3 Numero 2 >
Expectável : true | Recebido : " + coleccaoNivel3_2.addObra(livroTeste6));
            System.out.println("Adicionar Livro Teste 6 a Nivel 3 Numero 2 >
Expectável : false | Recebido : " + coleccaoNivel3_2.addObra(livroTeste6));
            System.out.println();

            System.out.println();
            System.out.println("Resultado Até Agora: ");
            coleccaoPrincipal.print("");
            System.out.println();

            System.out.println();
            System.out.println("################################       -xXx
addObra xXx- ################################");

            System.out.println();
            System.out.println();

            //getNumPaginas
            System.out.println("################################          -
getNumPaginas - ################################");
            System.out.println();

            System.out.println("getNumPaginas    Coleccao    Principal    >
Expectável : 369 | Recebido : " + coleccaoPrincipal.getNumPaginas());
            System.out.println("getNumPaginas Coleccao Nivel 2 Numero 1 >
Expectável : 119 | Recebido : " + coleccaoNivel2_1.getNumPaginas());
            System.out.println("getNumPaginas Coleccao Nivel 2 Numero 2 >
Expectável : 50  | Recebido : " + coleccaoNivel2_2.getNumPaginas());
            System.out.println("getNumPaginas Coleccao Nivel 3 Numero 1 >
Expectável : 50  | Recebido : " + coleccaoNivel3_1.getNumPaginas());
            System.out.println();

            System.out.println();
            System.out.println("Resultado Até Agora: ");
            coleccaoPrincipal.print("");
            System.out.println();

            System.out.println();
            System.out.println("################################       -xXx
getNumPaginas xXx- ################################");

            System.out.println();
            System.out.println();
```

```
            //getPreco
            System.out.println("################################## - getPreco
- ##################################");
            System.out.println();

            System.out.println("getNumPaginas    Coleccao    Principal    >
Expectável : 369 | Recebido : " + coleccaoPrincipal.getNumPaginas());
            System.out.println("getNumPaginas Coleccao Nivel 2 Numero 1 >
Expectável : 119 | Recebido : " + coleccaoNivel2_1.getNumPaginas());
            System.out.println("getNumPaginas Coleccao Nivel 2 Numero 2 >
Expectável : 50  | Recebido : " + coleccaoNivel2_2.getNumPaginas());
            System.out.println("getNumPaginas Coleccao Nivel 3 Numero 1 >
Expectável : 50  | Recebido : " + coleccaoNivel3_1.getNumPaginas());
            System.out.println();

            System.out.println();
            System.out.println("Resultado Até Agora: ");
            coleccaoPrincipal.print("");
            System.out.println();

            System.out.println();
            System.out.println("##################################       -xXx
getPreco xXx- ##################################");

            System.out.println();
            System.out.println();

            //getNumLivros
            System.out.println("##################################          -
getNumLivros - ##################################");
            System.out.println();

            System.out.println("getNumLivros    Coleccao    Principal    >
Expectável : 7 | Recebido : " + coleccaoPrincipal.getNumLivros());
            System.out.println("getNumLivros Coleccao Nivel 2 Numero 1 >
Expectável : 2 | Recebido : " + coleccaoNivel2_1.getNumLivros());
            System.out.println("getNumLivros Coleccao Nivel 2 Numero 2 >
Expectável : 1 | Recebido : " + coleccaoNivel2_2.getNumLivros());
            System.out.println("getNumLivros Coleccao Nivel 3 Numero 1 >
Expectável : 1 | Recebido : " + coleccaoNivel3_1.getNumLivros());
            System.out.println();

            System.out.println();
            System.out.println("Resultado Até Agora: ");
            coleccaoPrincipal.print("");
            System.out.println();

            System.out.println();
            System.out.println("##################################       -xXx
getNumLivros xXx- ##################################");

            System.out.println();
            System.out.println();
```

```
            //getNumColeccoes
            System.out.println("################################## -
getNumColeccoes - ##################################");
            System.out.println();

            System.out.println("getNumColeccoes Coleccao Principal >
Expectável : 4 | Recebido : " + coleccaoPrincipal.getNumColeccoes());
            System.out.println("getNumColeccoes Coleccao Nivel 2 Numero 1 >
Expectável : 0 | Recebido : " + coleccaoNivel2_1.getNumColeccoes());
            System.out.println("getNumColeccoes Coleccao Nivel 2 Numero 2 >
Expectável : 1 | Recebido : " + coleccaoNivel2_2.getNumColeccoes());
            System.out.println("getNumColeccoes Coleccao Nivel 3 Numero 1 >
Expectável : 0 | Recebido : " + coleccaoNivel3_1.getNumColeccoes());
            System.out.println();

            System.out.println();
            System.out.println("Resultado Até Agora: ");
            coleccaoPrincipal.print("");
            System.out.println();

            System.out.println();
            System.out.println("################################## -xXx
getNumColeccoes xXx- ##################################");

            System.out.println();
            System.out.println();

            //getIndexOfObra
            System.out.println("################################## -
getIndexOfObra - ##################################");
            System.out.println();

            System.out.println("getIndexObra Livro Teste 1 na Coleccao
Principal > Expectável : 0 | Recebido : " +
coleccaoPrincipal.getIndexOfObra("Livro Teste 1"));
            System.out.println("getIndexObra Coleccao Nivel 2 Numero 2 na
Coleccao Principal > Expectável : 2 | Recebido : " +
coleccaoPrincipal.getIndexOfObra("Coleccao Nivel 2 Numero 2"));
            System.out.println("getIndexObra Livro Teste 3 na Coleccao
Principal > Expectável : -1 | Recebido : " +
coleccaoPrincipal.getIndexOfObra("Livro Teste 3"));
            System.out.println("getIndexObra Livro Teste 3 na Coleccao Nivel
2 Numero 1 > Expectável : 1 | Recebido : " +
coleccaoNivel2_1.getIndexOfObra("Livro Teste 3"));
            System.out.println();

            System.out.println();
            System.out.println("Resultado Até Agora: ");
            coleccaoPrincipal.print("");
            System.out.println();

            System.out.println();
```

```
            System.out.println("################################        -xXx
getIndexOfObra xXx- ################################");

            System.out.println();
            System.out.println();

            //getNumObrasFromPerson
            System.out.println("################################          -
getNumObrasFromPerson - ################################");
            System.out.println();

            System.out.println("getNumObrasFromPerson    Afonso    Risco    na
Coleccao  Principal  >                    Expectável  :  4  |  Recebido  :  " +
coleccaoPrincipal.getNumObrasFromPerson("Afonso Risco"));
            System.out.println("getNumObrasFromPerson    Afonso    Risco    na
Coleccao  Nivel  2  Numero  2  >          Expectável  :  1  |  Recebido  :  " +
coleccaoNivel2_2.getNumObrasFromPerson("Afonso Risco"));
            System.out.println("getNumObrasFromPerson    Samuel    Torpedo    na
Coleccao  Nivel  3  Numero  1  >      Expectável  :  1  |  Recebido  :  " +
coleccaoNivel3_1.getNumObrasFromPerson("Samuel Torpedo"));
            System.out.println("getNumObrasFromPerson  Eduardo  Pedreiro  na
Coleccao  Nivel  3  Numero  1  >  Expectável  :  0  |  Recebido  :  " +
coleccaoNivel3_1.getNumObrasFromPerson("Eduardo Pedreiro"));
            System.out.println();

            System.out.println();
            System.out.println("Resultado Até Agora: ");
            coleccaoPrincipal.print("");
            System.out.println();

            System.out.println();
            System.out.println("################################        -xXx
getNumObrasFromPerson xXx- ################################");

            System.out.println();
            System.out.println();
```

```
            //getLivrosComoAutor
            System.out.println("################################## -
getLivrosComoAutor - ##################################");
            System.out.println();

            Livro[] livrosComoAutorTeste1 =
coleccaoPrincipal.getLivrosComoAutor("Afonso Risco");

            System.out.print("getLivrosComoAutor Afonso Risco na Coleccao
Principal > Expectável : [Livro Teste 1, Livro Teste 6] | Recebido : ");

            for(int livrosIndex = 0; livrosIndex <
livrosComoAutorTeste1.length; livrosIndex++)
            {
                  if(livrosIndex == 0)
                  {
                        System.out.print("[");
                  }


System.out.print(livrosComoAutorTeste1[livrosIndex].getTitulo());

                  if(livrosIndex == livrosComoAutorTeste1.length - 1)
                  {
                        System.out.println("]");
                  }
                  else
                  {
                        System.out.print(", ");
                  }
            }
```

```
            Livro[]                 livrosComoAutorTeste2                 =
coleccaoNivel2_2.getLivrosComoAutor("Afonso Risco");

            System.out.print("getLivrosComoAutor  Afonso  Risco  na  Coleccao
Nivel 2 Numero 2 > Expectável : [Livro Teste 1] | Recebido : ");

            for(int      livrosIndex      =      0;      livrosIndex      <
livrosComoAutorTeste2.length; livrosIndex++)
            {
                  if(livrosIndex == 0)
                  {
                        System.out.print("[");
                  }


System.out.print(livrosComoAutorTeste2[livrosIndex].getTitulo());

                  if(livrosIndex == livrosComoAutorTeste2.length - 1)
                  {
                        System.out.println("]");
                  }
                  else
                  {
                        System.out.print(", ");
                  }
            }

            Livro[]                 livrosComoAutorTeste3                 =
coleccaoPrincipal.getLivrosComoAutor("Bruno Aleixo");

            System.out.print("getLivrosComoAutor  Bruno  Aleixo  na  Coleccao
Principal > Expectável : | Recebido : ");

            for(int      livrosIndex      =      0;      livrosIndex      <
livrosComoAutorTeste3.length; livrosIndex++)
            {
                  if(livrosIndex == 0)
                  {
                        System.out.print("[");
                  }


System.out.print(livrosComoAutorTeste3[livrosIndex].getTitulo());

                  if(livrosIndex == livrosComoAutorTeste3.length - 1)
                  {
                        System.out.println("]");
                  }
                  else
                  {
                        System.out.print(", ");
                  }
            }
```

```
            System.out.println();

            Livro[]                  livrosComoAutorTeste4                  =
coleccaoPrincipal.getLivrosComoAutor("Vera Cruz");

            System.out.print("getLivrosComoAutor  Vera  Cruz  na  Coleccao
Principal > Expectável : [Livro Teste 6] | Recebido : ");

            for(int      livrosIndex      =      0;      livrosIndex      <
livrosComoAutorTeste4.length; livrosIndex++)
            {
                  if(livrosIndex == 0)
                  {
                        System.out.print("[");
                  }


System.out.print(livrosComoAutorTeste4[livrosIndex].getTitulo());

                  if(livrosIndex == livrosComoAutorTeste4.length - 1)
                  {
                        System.out.println("]");
                  }
                  else
                  {
                        System.out.print(", ");
                  }
            }
            System.out.println();

            System.out.println();
            System.out.println("Resultado Até Agora: ");
            coleccaoPrincipal.print("");
            System.out.println();

            System.out.println();
            System.out.println("##################################       -xXx
getLivrosComoAutor xXx- ##################################");

            System.out.println();
            System.out.println();
```

```
            //getAutoresEditores
            System.out.println("################################## -
getAutoresEditores - ##################################");
            System.out.println();

            String[]                  autoresEditoresTeste1                  =
coleccaoPrincipal.getAutoresEditores();

            System.out.println("getAutoresEditores da Coleccao Principal");
            System.out.println("Expectável : [Bruno Aleixo, Afonso Risco,
Telmo Afonso, Beatriz Turim, Gabriel Caparra, Filomena T Real, Carlos
Fonseca, José Ribeiro, Samuel Torpedo, Manuel Teixeira, Miguel Luz, Sara Luz,
Marco Luz, Xavier Terno, Telmo Guilherme, Lucinda Ferro, Diogo Ventura]");
            System.out.print("Recebido :   ");

            for(int  indexAutoresEditores  =  0;  indexAutoresEditores  <
autoresEditoresTeste1.length; indexAutoresEditores++)
            {
                  if(indexAutoresEditores == 0)
                  {
                        System.out.print("[");
                  }


System.out.print(autoresEditoresTeste1[indexAutoresEditores]);

                  if(indexAutoresEditores  ==  autoresEditoresTeste1.length  -
1)
                  {
                        System.out.println("]");
                  }
                  else
                  {
                        System.out.print(", ");
                  }
            }
            System.out.println();
```

```
            String[]              livrosAutoresEditoresTeste2              =
coleccaoNivel2_1.getAutoresEditores();

            System.out.println("getAutoresEditores  da  Coleccao  Nivel  2
Numero 1");
            System.out.println("Expectável  :  [Telmo  Afonso,  Beatriz  Turim,
Gabriel Caparra, Filomena T Real]");
            System.out.print("Recebido :   ");

            for(int  indexAutoresEditores  =  0;  indexAutoresEditores  <
livrosAutoresEditoresTeste2.length; indexAutoresEditores++)
            {
                  if(indexAutoresEditores == 0)
                  {
                        System.out.print("[");
                  }


System.out.print(livrosAutoresEditoresTeste2[indexAutoresEditores]);

                  if(indexAutoresEditores                                    ==
livrosAutoresEditoresTeste2.length - 1)
                  {
                        System.out.println("]");
                  }
                  else
                  {
                        System.out.print(", ");
                  }
            }
            System.out.println();
```

```
            String[]              livrosAutoresEditoresTeste3              =
coleccaoNivel2_2.getAutoresEditores();

            System.out.println("getAutoresEditores  da  Coleccao  Nivel  2
Numero 2");
            System.out.println("Expectável : [Carlos Fonseca, José Ribeiro,
Samuel Torpedo, Manuel Teixeira, Afonso Risco]");
            System.out.print("Recebido :   ");

            for(int  indexAutoresEditores  =  0;  indexAutoresEditores  <
livrosAutoresEditoresTeste3.length; indexAutoresEditores++)
            {
                  if(indexAutoresEditores == 0)
                  {
                        System.out.print("[");
                  }


System.out.print(livrosAutoresEditoresTeste3[indexAutoresEditores]);

                  if(indexAutoresEditores                                  ==
livrosAutoresEditoresTeste3.length - 1)
                  {
                        System.out.println("]");
                  }
                  else
                  {
                        System.out.print(", ");
                  }
            }
            System.out.println();
```

```
            String[]                livrosAutoresEditoresTeste4                =
coleccaoNivel3_1.getAutoresEditores();

            System.out.println("getAutoresEditores   da   Coleccao   Nivel   3
Numero 1");
            System.out.println("Expectável   :   [Samuel   Torpedo,   Manuel
Teixeira, Afonso Risco]");
            System.out.print("Recebido :   ");

            for(int   indexAutoresEditores   =   0;   indexAutoresEditores   <
livrosAutoresEditoresTeste4.length; indexAutoresEditores++)
            {
                  if(indexAutoresEditores == 0)
                  {
                        System.out.print("[");
                  }


System.out.print(livrosAutoresEditoresTeste4[indexAutoresEditores]);

                  if(indexAutoresEditores                                    ==
livrosAutoresEditoresTeste4.length - 1)
                  {
                        System.out.println("]");
                  }
                  else
                  {
                        System.out.print(", ");
                  }
            }
            System.out.println();
```

```
            String[]              livrosAutoresEditoresTeste5              =
coleccaoNivel3_2.getAutoresEditores();

            System.out.println("getAutoresEditores  da  Coleccao  Nivel  3
Numero 2");
            System.out.println("Expectável : [Afonso Risco, Vera Cruz]");
            System.out.print("Recebido :   ");

            for(int  indexAutoresEditores  =  0;  indexAutoresEditores  <
livrosAutoresEditoresTeste5.length; indexAutoresEditores++)
            {
                  if(indexAutoresEditores == 0)
                  {
                        System.out.print("[");
                  }


System.out.print(livrosAutoresEditoresTeste5[indexAutoresEditores]);

                  if(indexAutoresEditores                                  ==
livrosAutoresEditoresTeste5.length - 1)
                  {
                        System.out.println("]");
                  }
                  else
                  {
                        System.out.print(", ");
                  }
            }
            System.out.println();

            System.out.println();
            System.out.println("Resultado Até Agora: ");
            coleccaoPrincipal.print("");
            System.out.println();

            System.out.println();
            System.out.println("##################################       -xXx
getAutoresEditores xXx- ##################################");

            System.out.println();
            System.out.println();
```

```
            //equals
            System.out.println("################################## - equals -
##################################");
            System.out.println();

            Coleccao coleccaoEqualsTeste1_1 = new Coleccao("Coleccao Nivel 2
Numero 1", new String[] {"Telmo Afonso"});
            Coleccao coleccaoEqualsTeste1_2 = new Coleccao("Coleccao Nivel 2
Numero 1", new String[] {"Daniel Pereira"});
            Coleccao coleccaoEqualsTeste3_1 = new Coleccao("Coleccao Nivel 3
Numero 1", new String[] {"Samuel Torpedo", "Manuel Teixeira"});
            Coleccao coleccaoEqualsTeste3_2 = new Coleccao("Coleccao Nivel 3
Numero 2", new String[] {"Afonso Risco"});

            coleccaoEqualsTeste1_1.addObra(livroTeste1);
            coleccaoEqualsTeste1_1.addObra(coleccaoEqualsTeste3_1);

            coleccaoNivel2_1.print("");
            System.out.println();
            System.out.println("equals");
            System.out.println();
            coleccaoEqualsTeste1_1.print("");
            System.out.println();

System.out.println("-------------------------------------------------");
            System.out.println("Expectável : true | Recebido : " +
coleccaoNivel2_1.equals(coleccaoEqualsTeste1_1));

            System.out.println();

System.out.println("||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
");
            System.out.println();

            coleccaoEqualsTeste1_2.addObra(coleccaoEqualsTeste3_1);

            coleccaoNivel2_1.print("");
            System.out.println();
            System.out.println("equals");
            System.out.println();
            coleccaoEqualsTeste1_2.print("");
            System.out.println();

System.out.println("-------------------------------------------------");
            System.out.println("Expectável : false | Recebido : " +
coleccaoNivel2_1.equals(coleccaoEqualsTeste1_2));
            System.out.println();

            System.out.println();

System.out.println("||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
");
            System.out.println();
```

```
            coleccaoEqualsTeste3_2.addObra(livroTeste6);

            coleccaoNivel3_2.print("");
            System.out.println();
            System.out.println("equals");
            System.out.println();
            coleccaoEqualsTeste3_2.print("");
            System.out.println();

System.out.println("-------------------------------------------------");
            System.out.println("Expectável : true | Recebido : " +
coleccaoNivel3_2.equals(coleccaoEqualsTeste3_2));
            System.out.println();

            System.out.println();

System.out.println("||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
");
            System.out.println();

            System.out.println();
            System.out.println("Resultado Até Agora: ");
            coleccaoPrincipal.print("");
            System.out.println();

            System.out.println();
            System.out.println("##################################       -xXx
equals xXx- ##################################");

            System.out.println();
            System.out.println();
```

```
            //remObra
            System.out.println("################################## - remObra
- ##################################");
            System.out.println();

            System.out.println("Para testes, criação de Livro e Coleccao com
o nome: Remover Teste");

            Livro removerLivroTeste = new Livro("Remover Teste", 10, 10f,
new String[] {"Autor Teste"});
            Coleccao removerColeccaoTeste = new Coleccao("Remover Teste",
new String[] {"Autor Teste"});

            coleccaoPrincipal.addObra(removerLivroTeste);
            coleccaoNivel3_2.addObra(removerColeccaoTeste);

            System.out.println();
            System.out.println("Resultado: ");
            coleccaoPrincipal.print("");
            System.out.println();

            System.out.println("Remover Remover Teste da Coleccao
Principal");
            System.out.println("Expectável : Remover Teste | Obra Removida :
" + coleccaoPrincipal.remObra("Remover Teste"));
            System.out.println();
            System.out.println("||||||||||||||||||");
            System.out.println();
            System.out.println("Resultado: ");
            coleccaoPrincipal.print("");
            System.out.println();

            System.out.println();
            System.out.println("-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-");
            System.out.println();

            System.out.println("Remover Remover Teste da Coleccao
Principal");
            System.out.println("Expectável : null | Obra Removida : " +
coleccaoPrincipal.remObra("Remover Teste"));
            System.out.println();
            System.out.println("||||||||||||||||||");
            System.out.println();
            System.out.println("Resultado: ");
            coleccaoPrincipal.print("");
            System.out.println();

            System.out.println();
            System.out.println("-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-");
            System.out.println();
```

```
            System.out.println("Remover Remover Teste da Coleccao Nivel 3
Numero 2");
            System.out.println("Expectável : Remover Teste | Obra Removida :
" + coleccaoNivel3_2.remObra("Remover Teste"));
            System.out.println();
            System.out.println("||||||||||||||||||");
            System.out.println();
            System.out.println("Resultado: ");
            coleccaoPrincipal.print("");
            System.out.println();

            System.out.println();
            System.out.println("################################        -xXx
remObra xXx- ################################");

            System.out.println();
            System.out.println();

            //remAllObra
            System.out.println("################################          -
remAllObra - ################################");
            System.out.println();

            System.out.println("Para testes, criação de Livro e Coleccao com
o nome: Remover Teste");

            coleccaoPrincipal.addObra(removerLivroTeste);
            coleccaoNivel3_2.addObra(removerColeccaoTeste);

            System.out.println();
            System.out.println("Resultado: ");
            coleccaoPrincipal.print("");
            System.out.println();

            System.out.println("Remover Todos Remover Teste da Coleccao
Principal");
            System.out.println("Expectável : true | Recebido : " +
coleccaoPrincipal.remAllObra("Remover Teste"));
            System.out.println();
            System.out.println("||||||||||||||||||");
            System.out.println();
            System.out.println("Resultado: ");
            coleccaoPrincipal.print("");
            System.out.println();

            System.out.println();
            System.out.println("-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-");
            System.out.println();
```

```
            System.out.println("Remover Todos Remover Teste da Coleccao
Principal");
            System.out.println("Expectável : false | Recebido : " +
coleccaoPrincipal.remAllObra("Remover Teste"));
            System.out.println();
            System.out.println("||||||||||||||||");
            System.out.println();
            System.out.println("Resultado: ");
            coleccaoPrincipal.print("");
            System.out.println();

            System.out.println();
            System.out.println("################################       -xXx
remAllObra xXx- ################################");
      }
}
```

2. pack2Festivais

2.1. Evento

Nesta classe abstracta, existem três métodos abstractos: *getNumBilhetes*, *getArtistas* e *numActuacoes*.

A classe possui um constructor que recebe uma *String*. Esta *String* não pode ser *null*, não pode ser vazia e tem de possuir pelo menos um caractere. Só pode ter letras, espaços e números.

O método *toString* devolve uma com *String* que possui o nome do Evento, o número de bilhetes e os seus artistas.

Código:

```
package tp2.pack2Festivais;

public abstract class Evento {

	private String nome;

	public Evento(String nome)
	{
		if(nome == null)
		{
			throw new IllegalArgumentException("O nome não pode ser
null");
		}

		if(nome.length() == 0)
		{
			throw new IllegalArgumentException("O tamanho do nome não
pode ser 0");
		}

		int letterCounter = 0;

		for(int charIndex = 0; charIndex < nome.length(); charIndex++)
	{
		if(!Character.isLetter(nome.charAt(charIndex))                &&
!Character.isWhitespace(nome.charAt(charIndex))                       &&
!Character.isDigit(nome.charAt(charIndex)))
			{
			throw new IllegalArgumentException("O nome só pode possuir
letras, espaços e números");
			}

		if(Character.isLetter(nome.charAt(charIndex)))
		{
			letterCounter++;
		}
	}

		if(letterCounter == 0)
		{
			throw new IllegalArgumentException("O nome tem de ter,
pelo menos, um caractere");
		}
```

```
            this.nome = nome;
      }

      public abstract int getNumBilhetes();

      public abstract String[] getArtistas();

      public abstract int numActuacoes(String artista);

      public String toString()
      {
            String toString = nome + " com " + getNumBilhetes() + " bilhetes
e com ";

            if(getArtistas().length > 1)
            {
                  toString += "os artistas [";
            }
            else
            {
                  toString += "o artista [";
            }

            for(int index = 0; index < getArtistas().length; index++)
            {
                  toString += getArtistas()[index];

                  if(index != getArtistas().length - 1)
                  {
                        toString += ", ";
                  }
            }

            toString += "]";

            return toString;
      }
}
```

2.2. Espectaculo

No constructor da classe Espectaculo, o título passa pelo constructor da classe pai usando a *keyword*: *super*. Recebe uma *String* para a localidade e um *int* para o número de bilhetes. A localidade não pode ser *null*, vazia, tem de possuir pelo menos um caractere e só pode ser constituída por letras, espaços e números. O número de bilhetes não pode ser negativo.

numActuacoes recebe um artista e devolve o número de aparições que esse artista faz no *array* de artistas. *addArtista* recebe uma *String* com o nome do artista mas este não é aceite se já existir no *array* de artistas.

getNumBilhetes devolve o número de bilhetes do Espectáculo. *getArtistas* devolve um *array* de *String*'s apenas com os artistas existentes no espectáculo.

toString devolve uma *String* que possui o *toString* da classe-pai, adicionando posteriormente a localidade.

Finalmente, possui uma classe *main* onde são efectuados diversos testes a todos os métodos da classe.

Código:

```
package tp2.pack2Festivais;

public class Espectaculo extends Evento{

	private int nArtistas = 0;

	private String[] artistas = new String[10];

	private int numBilhetes = 0;

	private String localidade = "";

	public Espectaculo(String nome, String localidade, int numBilhetes)
	{
		super(nome);

		//Localidade
		if(localidade == null)
		{
			throw new IllegalArgumentException("A localidade não pode
ser null");
		}

		if(localidade.length() == 0)
		{
			throw new IllegalArgumentException("O tamanho da
localidade não pode ser 0");
		}

		int letterCounter = 0;
```

```
            for(int charIndex = 0; charIndex < localidade.length();
charIndex++)
      {
            if(!Character.isLetter(localidade.charAt(charIndex))          &&
!Character.isWhitespace(localidade.charAt(charIndex))                     &&
!Character.isDigit(localidade.charAt(charIndex)))
                  {
                  throw new IllegalArgumentException("A localidade só pode
possuir letras, espaços e números");
                  }

            if(Character.isLetter(localidade.charAt(charIndex)))
            {
                  letterCounter++;
            }
      }

            if(letterCounter == 0)
            {
                  throw new IllegalArgumentException("A localidade tem de
ter, pelo menos, um caractere");
            }

            this.localidade = localidade;

            //NumBilhetes
            if(numBilhetes <= 0)
            {
                  throw new IllegalArgumentException("O número de bilhetes
deve ser superior a 0");
            }

            this.numBilhetes = numBilhetes;
      }

      public int numActuacoes(String artista) {

            for(int index = 0; index < nArtistas; index++)
            {
                  if(artistas[index].equals(artista))
                  {
                        return 1;
                  }
            }

            return 0;
      }
```

```
public boolean addArtista(String artista)
{
	if(numActuacoes(artista) == 1 || nArtistas == artistas.length)
	{
		return false;
	}

	artistas[nArtistas] = artista;
	nArtistas++;

	return true;
}

public int getNumBilhetes() {

	return numBilhetes;
}

public String[] getArtistas() {

	String[] artistasRegistados = new String[nArtistas];

	for(int index = 0; index < nArtistas; index++)
	{
		artistasRegistados[index] = artistas[index];
	}

	return artistasRegistados;
}

public String toString()
{
	return super.toString() + " em " + localidade;
}

public static void main(String[] args) {

	System.out.println("#####---- Testes ----#####");
	System.out.println();

	System.out.println(" - Testar Constructor - ");
	System.out.println();

	System.out.println("####### Nome #######");
	System.out.println();
```

```
		System.out.println(" - Nome é null");
		try {
			Espectaculo espectaculoErrado = new Espectaculo(null,
"Main", 404);
		}
		catch(IllegalArgumentException ex)
		{
			ex.printStackTrace();
		}
		System.out.println();

		System.out.println(" - Nome é vazio");
		try {
			Espectaculo espectaculoErrado = new Espectaculo("",
"Main", 404);
		}
		catch(IllegalArgumentException ex)
		{
			ex.printStackTrace();
		}
		System.out.println();

		System.out.println(" - Nome não pode possuir simbolos");
		try {
			Espectaculo espectaculoErrado = new Espectaculo("Nome
Inválido #", "Main", 404);
		}
		catch(IllegalArgumentException ex)
		{
			ex.printStackTrace();
		}
		System.out.println();

		System.out.println(" - Nome Correcto");
		try {
			Espectaculo espectaculoErrado = new Espectaculo("Nome
Válido 23245     F", "Main", 404);
			System.out.println(espectaculoErrado.toString());
		}
		catch(IllegalArgumentException ex)
		{
			ex.printStackTrace();
		}
		System.out.println();

		System.out.println("####### Nome #######");
		System.out.println();

		System.out.println("####### Localidade #######");
		System.out.println();
```

```
            System.out.println(" - Localidade é null");
            try {
                  Espectaculo  espectaculoErrado  =  new  Espectaculo("Nome
Válido", null, 404);
            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }
            System.out.println();

            System.out.println(" - Localidade é vazio");
            try {
                  Espectaculo  espectaculoErrado  =  new  Espectaculo("Nome
Válido", "", 404);
            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }
            System.out.println();

            System.out.println(" - Localidade não pode possuir simbolos");
            try {
                  Espectaculo  espectaculoErrado  =  new  Espectaculo("Nome
Válido", "Main #", 404);
            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }
            System.out.println();

            System.out.println(" - Localidade Correcta");
            try {
                  Espectaculo  espectaculoErrado  =  new  Espectaculo("Nome
Válido", "Eclipse Main", 404);
                  System.out.println(espectaculoErrado.toString());
            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }
            System.out.println();

            System.out.println("####### Localidade #######");
            System.out.println();

            System.out.println("####### Número de Bilhetes #######");
            System.out.println();
```

```
            System.out.println(" - numBilhetes é negativo");
            try {
                  Espectaculo  espectaculoErrado  =  new  Espectaculo("Nome
Válido", "Eclipse Main", -5);
            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }
            System.out.println();

            System.out.println(" - numBilhetes é 0");
            try {
                  Espectaculo  espectaculoErrado  =  new  Espectaculo("Nome
Válido", "Eclipse Main", 0);
            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }
            System.out.println();

            System.out.println(" - numBilhetes Correcto");
            try {
                  Espectaculo  espectaculoErrado  =  new  Espectaculo("Nome
Válido", "Eclipse Main", 100);
                  System.out.println(espectaculoErrado.toString());
            }
            catch(IllegalArgumentException ex)
            {
                  ex.printStackTrace();
            }
            System.out.println();

            System.out.println("####### Numero de Bilhetes #######");
            System.out.println();

            System.out.println(" - Testar Métodos - ");
            System.out.println();

            Espectaculo espectaculoTeste1 = new Espectaculo("Teste Espectulo
1", "Eclipse Main", 120);
            Espectaculo espectaculoTeste2 = new Espectaculo("Teste Espectulo
2", "ISEL", 200);
            Espectaculo espectaculoTeste3 = new Espectaculo("Teste Espectulo
3", "Marte", 1250);
```

```
            System.out.println("Espectáculos: ");
            System.out.println(espectaculoTeste1.toString());
            System.out.println(espectaculoTeste2.toString());
            System.out.println(espectaculoTeste3.toString());
            System.out.println();

            System.out.println("####### getNumBilhetes #######");
            System.out.println();

            System.out.println("Espectaculo Teste 1 > Expectável : 120 |
Recebido : " + espectaculoTeste1.getNumBilhetes());
            System.out.println("Espectaculo Teste 2 > Expectável : 200 |
Recebido : " + espectaculoTeste2.getNumBilhetes());
            System.out.println("Espectaculo Teste 3 > Expectável : 1250 |
Recebido : " + espectaculoTeste3.getNumBilhetes());
            System.out.println();

            System.out.println("####### getNumBilhetes #######");
            System.out.println();

            System.out.println("####### addArtista #######");
            System.out.println();

            String artistaTeste1 = "Afonso Risco";
            String artistaTeste2 = "Beatriz Valeta";
            String artistaTeste3 = "Carlos Trindade";
            String artistaTeste4 = "Dionisio Asa";
            String artistaTeste5 = "Eduardo Carreira";
            String artistaTeste6 = "Francisca Gilberto";

            System.out.println("Adicionar Afonso Risco, duas vezes, a
Espectaculo Teste 1");
            System.out.println("Adicionar Afonso Risco a Espectaculo Teste 1
>     Expectável     :     true          |     Recebido     :     "     +
espectaculoTeste1.addArtista(artistaTeste1));
            System.out.println("Adicionar Afonso Risco a Espectaculo Teste 1
>     Expectável     :     false     |     Recebido     :     "     +
espectaculoTeste1.addArtista(artistaTeste1));
            System.out.println();

            System.out.println("Adicionar Beatriz Valeta, duas vezes, a
Espectaculo Teste 2");
            System.out.println("Adicionar Beatriz Valeta a Espectaculo Teste
2     >     Expectável     :     true          |     Recebido     :     "     +
espectaculoTeste2.addArtista(artistaTeste2));
            System.out.println("Adicionar Beatriz Valeta a Espectaculo Teste
2     >     Expectável     :     false     |     Recebido     :     "     +
espectaculoTeste2.addArtista(artistaTeste2));
            System.out.println();
```

```
            System.out.println("Adicionar Carlos Trindade, duas vezes, a
Espectaculo Teste 2");
            System.out.println("Adicionar Carlos Trindade a Espectaculo
Teste 2 > Expectável : true | Recebido : " +
espectaculoTeste2.addArtista(artistaTeste3));
            System.out.println("Adicionar Carlos Trindade a Espectaculo
Teste 2 > Expectável : false | Recebido : " +
espectaculoTeste2.addArtista(artistaTeste3));
            System.out.println();

            System.out.println("Adicionar Dionisio Asa, duas vezes, a
Espectaculo Teste 3");
            System.out.println("Adicionar Dionisio Asa a Espectaculo Teste 3
> Expectável : true | Recebido : " +
espectaculoTeste3.addArtista(artistaTeste4));
            System.out.println("Adicionar Dionisio Asa a Espectaculo Teste 3
> Expectável : false | Recebido : " +
espectaculoTeste3.addArtista(artistaTeste4));
            System.out.println();

            System.out.println("Adicionar Eduardo Carreira, duas vezes, a
Espectaculo Teste 3");
            System.out.println("Adicionar Eduardo Carreira a Espectaculo
Teste 3 > Expectável : true | Recebido : " +
espectaculoTeste3.addArtista(artistaTeste5));
            System.out.println("Adicionar Eduardo Carreira a Espectaculo
Teste 3 > Expectável : false | Recebido : " +
espectaculoTeste3.addArtista(artistaTeste5));
            System.out.println();

            System.out.println("Adicionar Francisca Gilberto, duas vezes, a
Espectaculo Teste 3");
            System.out.println("Adicionar Francisca Gilberto a Espectaculo
Teste 3 > Expectável : true | Recebido : " +
espectaculoTeste3.addArtista(artistaTeste6));
            System.out.println("Adicionar Francisca Gilberto a Espectaculo
Teste 3 > Expectável : false | Recebido : " +
espectaculoTeste3.addArtista(artistaTeste6));
            System.out.println();

            System.out.println("Espectáculos: ");
            System.out.println(espectaculoTeste1.toString());
            System.out.println(espectaculoTeste2.toString());
            System.out.println(espectaculoTeste3.toString());
            System.out.println();

            System.out.println("####### addArtista #######");
            System.out.println();
```

```
            System.out.println("####### getArtistas #######");
            System.out.println();

            System.out.print("Artistas do Espectaculo Teste 1 : ");
            String[]                artistasEspectaculoTeste1                =
espectaculoTeste1.getArtistas();

            for(int      indexArtista      =      0;      indexArtista      <
artistasEspectaculoTeste1.length; indexArtista++)
            {
                  if(indexArtista == 0)
                  {
                        System.out.print("[");
                  }

                  System.out.print(artistasEspectaculoTeste1[indexArtista]);

                  if(indexArtista == artistasEspectaculoTeste1.length - 1)
                  {
                        System.out.println("]");
                  }
                  else
                  {
                        System.out.print(", ");
                  }
            }
            System.out.println();
```

```
            System.out.print("Artistas do Espectaculo Teste 2 : ");
            String[]                artistasEspectaculoTeste2                =
espectaculoTeste2.getArtistas();

            for(int      indexArtista      =      0;      indexArtista      <
artistasEspectaculoTeste2.length; indexArtista++)
            {
                  if(indexArtista == 0)
                  {
                        System.out.print("[");
                  }

                  System.out.print(artistasEspectaculoTeste2[indexArtista]);

                  if(indexArtista == artistasEspectaculoTeste2.length - 1)
                  {
                        System.out.println("]");
                  }
                  else
                  {
                        System.out.print(", ");
                  }
            }
            System.out.println();

            System.out.print("Artistas do Espectaculo Teste 3 : ");
            String[]                artistasEspectaculoTeste3                =
espectaculoTeste3.getArtistas();

            for(int      indexArtista      =      0;      indexArtista      <
artistasEspectaculoTeste3.length; indexArtista++)
            {
                  if(indexArtista == 0)
                  {
                        System.out.print("[");
                  }

                  System.out.print(artistasEspectaculoTeste3[indexArtista]);

                  if(indexArtista == artistasEspectaculoTeste3.length - 1)
                  {
                        System.out.println("]");
                  }
                  else
                  {
                        System.out.print(", ");
                  }
            }
            System.out.println();

            System.out.println("####### getArtistas #######");
            System.out.println();
```

```
            System.out.println("####### numAtuacoes #######");
            System.out.println();

            System.out.println("Numero de Atuações de Afonso Risco no
Espectaculo Teste 1 > Expectável : 1 | Recebido : " +
espectaculoTeste1.numActuacoes(artistaTeste1));
            System.out.println("Numero de Atuações de Afonso Risco no
Espectaculo Teste 3 > Expectável : 0 | Recebido : " +
espectaculoTeste3.numActuacoes(artistaTeste1));
            System.out.println();

            System.out.println("Numero de Atuações de Beatriz Valeta no
Espectaculo Teste 2 > Expectável : 1 | Recebido : " +
espectaculoTeste2.numActuacoes(artistaTeste2));
            System.out.println("Numero de Atuações de Beatriz Valeta no
Espectaculo Teste 1 > Expectável : 0 | Recebido : " +
espectaculoTeste1.numActuacoes(artistaTeste2));
            System.out.println();

            System.out.println("Numero de Atuações de Eduardo Carreira no
Espectaculo Teste 3 > Expectável : 1 | Recebido : " +
espectaculoTeste3.numActuacoes(artistaTeste5));
            System.out.println("Numero de Atuações de Eduardo Carreira no
Espectaculo Teste 2 > Expectável : 0 | Recebido : " +
espectaculoTeste2.numActuacoes(artistaTeste5));
            System.out.println();

            System.out.println("####### numAtuacoes #######");
            System.out.println();
      }
}
```

2.3. Festival

Na classe Festival, existe um constructor que recebe uma *String* que é o nome e que é passada para o constructor da classe-pai, Evento.

getNumBilhetes devolve o número de bilhetes que o Festival possui. O método *numActuacoes* devolve o número de aparições do artista, recebido como argumento, no Festival.

O método *toString* devolve uma *String* que possui o nome do Festival, seguindo-se do *toString* da classe-pai. *getArtistas* devolve o *array* de *String*'s com todos os artistas do Festival.

getDeepFestival devolve a profundidade máxima que o Festival possui.

A função *addEvento*, recebe como argumento um Evento, e verifica se o número de atuações dos artistas em comum, com os dois Eventos, não ultrapassa o do evento a que este está a ser adicionado, mais dois. A função *delEvento* por sua vez, recebe o nome do Evento a remover e procura todos os Eventos correspondentes e remove-os do *array* de Eventos.

Por fim, temos o método *main* onde são realizados inúmeros testes a cada função da classe Festival.

Código:

```
package tp2.pack2Festivais;

import java.util.ArrayList;
import java.util.List;

public class Festival extends Evento {

	private int numEventos = 0;

	private Evento[] eventos = new Evento[20];

	public Festival(String nome) {
		super(nome);
	}
```

```
	public int getNumBilhetes() {

		int numBilhetes = 0;

		for(int index = 0; index < eventos.length; index++)
		{
			if(eventos[index] != null)
			{
				numBilhetes += eventos[index].getNumBilhetes();

			}
		}

		return numBilhetes;
	}

	public int numActuacoes(String artista) {

		int numActuacoes = 0;

		for(int index = 0; index < eventos.length; index++)
		{
			if(eventos[index] != null)
			{
				numActuacoes                                    +=
eventos[index].numActuacoes(artista);
			}
		}

		return numActuacoes;
	}

	public String toString()
	{
		return "Festival " + super.toString();
	}
```

```
	public String[] getArtistas() {

		List<String> artistas = new ArrayList<String>();

		for(int index = 0; index < eventos.length; index++)
		{
			if(eventos[index] != null)
			{
				String[] artistasRecolhidos = eventos[index].getArtistas();

				for(int indexArtistas = 0; indexArtistas < artistasRecolhidos.length; indexArtistas++)
				{

if(!artistas.contains(artistasRecolhidos[indexArtistas]))
					{

artistas.add(artistasRecolhidos[indexArtistas]);
					}
				}
			}
		}

		return (String[]) artistas.toArray(new String[artistas.size()]);
	}

	private int getDeepFestival()
	{
		int profundidade = -1;

		for(int index = 0; index < eventos.length; index++)
		{
			if(eventos[index] instanceof Festival)
			{
				profundidade = Math.max(profundidade, ((Festival) eventos[index]).getDeepFestival());
			}
		}

		profundidade++;

		return profundidade;
	}
```

```
	public boolean addEvento(Evento evento)
	{
		if(numEventos == eventos.length || evento == null)
		{
			return false;
		}

		if(evento.getArtistas().length == 0)
		{
			return false;
		}

		String[] artistasEvento = evento.getArtistas();

		for(int indexArtista = 0; indexArtista < getArtistas().length;
indexArtista++)
		{
			String artista = getArtistas()[indexArtista];

			for(int indexAComparar = 0; indexAComparar <
artistasEvento.length; indexAComparar++)
			{
				String artistaAComparar =
artistasEvento[indexAComparar];

				if(artista.equals(artistaAComparar))
				{
					if(evento.numActuacoes(artistaAComparar) >
numActuacoes(artista) + 2)
					{
						return false;
					}
				}
			}
		}

		for(int index = 0; index < eventos.length; index++)
		{
			if(eventos[index] == null)
			{
				eventos[index] = evento;
				numEventos += 1;
				break;
			}
		}

		return true;
	}
```

```
	public boolean delEvento(String nomeEvento)
	{
		if(numEventos == 0 || nomeEvento == null)
		{
			return false;
		}

		boolean removido = false;

		for(int index = 0; index < eventos.length; index++)
		{
			if(eventos[index] instanceof Espectaculo)
			{
				if(eventos[index].toString().contains(nomeEvento))
				{
					eventos[index] = null;
					numEventos--;
					removido = true;
				}
			}
			else if(eventos[index] instanceof Festival)
			{
				if(eventos[index].toString().contains(nomeEvento))
				{
					eventos[index] = null;
					numEventos--;
					removido = true;
				}
				else
				{
					boolean removidoRecursivo = ((Festival) eventos[index]).delEvento(nomeEvento);

					if(!removido && removidoRecursivo)
					{
						removido = removidoRecursivo;
					}
				}
			}
		}

		return removido;
	}
```

```
	public static void main(String[] args) {

		System.out.println("#####---- Testes ----#####");
		System.out.println();

		System.out.println(" - Testar Métodos - ");
		System.out.println();

		String artistaTeste1 = "Afonso Risco";
		String artistaTeste2 = "Beatriz Valeta";
		String artistaTeste3 = "Carlos Trindade";
		String artistaTeste4 = "Dionisio Asa";
		String artistaTeste5 = "Eduardo Carreira";
		String artistaTeste6 = "Francisca Gilberto";

		Espectaculo espectaculoTeste1 = new Espectaculo("Teste
Espectaculo 1", "Eclipse Main", 120);
		Espectaculo espectaculoTeste2 = new Espectaculo("Teste
Espectaculo 2", "ISEL", 200);
		Espectaculo espectaculoTeste3 = new Espectaculo("Teste
Espectaculo 3", "Marte", 1250);

		Festival festivalTeste1 = new Festival("Teste 1");
		Festival festivalTeste2 = new Festival("Teste 2");
		Festival festivalTeste3 = new Festival("Teste 3");
		Festival festivalTeste4 = new Festival("Teste 4");
		Festival festivalTeste5 = new Festival("Teste 5");

		espectaculoTeste1.addArtista(artistaTeste1);
		espectaculoTeste2.addArtista(artistaTeste2);
		espectaculoTeste2.addArtista(artistaTeste3);
		espectaculoTeste3.addArtista(artistaTeste4);
		espectaculoTeste3.addArtista(artistaTeste5);
		espectaculoTeste3.addArtista(artistaTeste6);

		System.out.println("Artistas: ");
		System.out.println(artistaTeste1);
		System.out.println(artistaTeste2);
		System.out.println(artistaTeste3);
		System.out.println(artistaTeste4);
		System.out.println(artistaTeste5);
		System.out.println(artistaTeste6);
		System.out.println();

		System.out.println("Espectáculos: ");
		System.out.println(espectaculoTeste1.toString());
		System.out.println(espectaculoTeste2.toString());
		System.out.println(espectaculoTeste3.toString());
		System.out.println();
```

```
            System.out.println("Festivais: ");
            System.out.println(festivalTeste1.toString());
            System.out.println(festivalTeste2.toString());
            System.out.println(festivalTeste3.toString());
            System.out.println(festivalTeste4.toString());
            System.out.println();

            System.out.println("####### addEvento #######");
            System.out.println();

            System.out.println("Adicionar Teste Espectaculo 1, quatro vezes,
a Festival Teste 1");
            System.out.println("Adicionar  Teste  Espectaculo  1  a  Festival
Teste   1   >   Expectável   :   true        |   Recebido   :   "   +
festivalTeste1.addEvento(espectaculoTeste1));
            System.out.println("Adicionar  Teste  Espectaculo  1  a  Festival
Teste   1   >   Expectável   :   true        |   Recebido   :   "   +
festivalTeste1.addEvento(espectaculoTeste1));
            System.out.println("Adicionar  Teste  Espectaculo  1  a  Festival
Teste   1   >   Expectável   :   true        |   Recebido   :   "   +
festivalTeste1.addEvento(espectaculoTeste1));
            System.out.println("Adicionar  Teste  Espectaculo  1  a  Festival
Teste   1   >   Expectável   :   true        |   Recebido   :   "   +
festivalTeste1.addEvento(espectaculoTeste1));
            System.out.println();

            System.out.println("Adicionar  Teste  Espectaculo  1  a  Festival
Teste 2");
            System.out.println("Adicionar  Teste  Espectaculo  1  a  Festival
Teste   2   >   Expectável   :   true        |   Recebido   :   "   +
festivalTeste2.addEvento(espectaculoTeste1));
            System.out.println();

            System.out.println("Adicionar Festival Teste 1 a Festival Teste
2");
            System.out.println("Adicionar Festival Teste 1 a Festival Teste
2   >   Expectável   :   false        |   Recebido   :   "   +
festivalTeste2.addEvento(festivalTeste1));
            System.out.println("Justificação : NumActuações de Afonso Risco
> Festival Teste 1 : " + festivalTeste1.numActuacoes(artistaTeste1) + " |
Festival Teste 2 : " + festivalTeste2.numActuacoes(artistaTeste1));
            System.out.println();

            System.out.println("Adicionar Teste Espectaculo 2, duas vezes, a
Festival Teste 3");
            System.out.println("Adicionar  Teste  Espectaculo  2  a  Festival
Teste   3   >   Expectável   :   true        |   Recebido   :   "   +
festivalTeste3.addEvento(espectaculoTeste2));
            System.out.println("Adicionar  Teste  Espectaculo  2  a  Festival
Teste   3   >   Expectável   :   true        |   Recebido   :   "   +
festivalTeste3.addEvento(espectaculoTeste2));
            System.out.println();
```

```
            System.out.println("Adicionar Teste Espectaculo 3, duas vezes, a
Festival Teste 3");
            System.out.println("Adicionar Teste Espectaculo 3 a Festival
Teste 3 > Expectável : true | Recebido : " +
festivalTeste3.addEvento(espectaculoTeste3));
            System.out.println("Adicionar Teste Espectaculo 3 a Festival
Teste 3 > Expectável : true | Recebido : " +
festivalTeste3.addEvento(espectaculoTeste3));
            System.out.println();

            System.out.println("Adicionar Festival Teste 1 a Festival Teste
4 > Expectável : true | Recebido : " +
festivalTeste4.addEvento(festivalTeste1));
            System.out.println("Adicionar Festival Teste 2 a Festival Teste
4 > Expectável : true | Recebido : " +
festivalTeste4.addEvento(festivalTeste2));
            System.out.println("Adicionar Festival Teste 3 a Festival Teste
4 > Expectável : true | Recebido : " +
festivalTeste4.addEvento(festivalTeste3));
            System.out.println();

            System.out.println("Adicionar Festival Teste 4 a Festival Teste
5 > Expectável : true | Recebido : " +
festivalTeste5.addEvento(festivalTeste4));
            System.out.println("Adicionar Festival Teste 1 a Festival Teste
5 > Expectável : true | Recebido : " +
festivalTeste5.addEvento(festivalTeste1));
            System.out.println("Adicionar Espectaculo Teste 1 a Festival
Teste 5 > Expectável : true | Recebido : " +
festivalTeste5.addEvento(espectaculoTeste1));
            System.out.println();

            System.out.println("Festivais: ");
            System.out.println(festivalTeste1.toString());
            System.out.println(festivalTeste2.toString());
            System.out.println(festivalTeste3.toString());
            System.out.println(festivalTeste4.toString());
            System.out.println(festivalTeste5.toString());
            System.out.println();

            System.out.println("####### addEvento #######");
            System.out.println();
```

```
		System.out.println("####### getArtistas #######");
		System.out.println();

		System.out.print("Artistas do Festival Teste 1 : ");
		String[] artistasFestivalTeste1 = festivalTeste1.getArtistas();

		for(int indexArtista = 0; indexArtista < artistasFestivalTeste1.length; indexArtista++)
		{
			if(indexArtista == 0)
			{
				System.out.print("[");
			}

			System.out.print(artistasFestivalTeste1[indexArtista]);

			if(indexArtista == artistasFestivalTeste1.length - 1)
			{
				System.out.println("]");
			}
			else
			{
				System.out.print(", ");
			}
		}
		System.out.println();

		System.out.print("Artistas do Festival Teste 2 : ");
		String[] artistasFestivalTeste2 = festivalTeste2.getArtistas();

		for(int indexArtista = 0; indexArtista < artistasFestivalTeste2.length; indexArtista++)
		{
			if(indexArtista == 0)
			{
				System.out.print("[");
			}

			System.out.print(artistasFestivalTeste2[indexArtista]);

			if(indexArtista == artistasFestivalTeste2.length - 1)
			{
				System.out.println("]");
			}
			else
			{
				System.out.print(", ");
			}
		}
		System.out.println();
```

```
            System.out.print("Artistas do Festival Teste 3 : ");
            String[] artistasFestivalTeste3 = festivalTeste3.getArtistas();

            for(int indexArtista = 0; indexArtista <
artistasFestivalTeste3.length; indexArtista++)
            {
                  if(indexArtista == 0)
                  {
                        System.out.print("[");
                  }

                  System.out.print(artistasFestivalTeste3[indexArtista]);

                  if(indexArtista == artistasFestivalTeste3.length - 1)
                  {
                        System.out.println("]");
                  }
                  else
                  {
                        System.out.print(", ");
                  }
            }
            System.out.println();

            System.out.print("Artistas do Festival Teste 4 : ");
            String[] artistasFestivalTeste4 = festivalTeste4.getArtistas();

            for(int indexArtista = 0; indexArtista <
artistasFestivalTeste4.length; indexArtista++)
            {
                  if(indexArtista == 0)
                  {
                        System.out.print("[");
                  }

                  System.out.print(artistasFestivalTeste4[indexArtista]);

                  if(indexArtista == artistasFestivalTeste4.length - 1)
                  {
                        System.out.println("]");
                  }
                  else
                  {
                        System.out.print(", ");
                  }
            }
            System.out.println();
```

```
            System.out.print("Artistas do Festival Teste 5 : ");
            String[] artistasFestivalTeste5 = festivalTeste5.getArtistas();

            for(int indexArtista = 0; indexArtista < artistasFestivalTeste5.length; indexArtista++)
            {
                  if(indexArtista == 0)
                  {
                        System.out.print("[");
                  }

                  System.out.print(artistasFestivalTeste5[indexArtista]);

                  if(indexArtista == artistasFestivalTeste5.length - 1)
                  {
                        System.out.println("]");
                  }
                  else
                  {
                        System.out.print(", ");
                  }
            }
            System.out.println();

            System.out.println("####### getArtistas #######");
            System.out.println();

            System.out.println("####### getnumBilhetes #######");
            System.out.println();

            System.out.println("Bilhetes do Festival Teste 1 : " + festivalTeste1.getNumBilhetes());
            System.out.println("Bilhetes do Festival Teste 2 : " + festivalTeste2.getNumBilhetes());
            System.out.println("Bilhetes do Festival Teste 3 : " + festivalTeste3.getNumBilhetes());
            System.out.println("Bilhetes do Festival Teste 4 : " + festivalTeste4.getNumBilhetes());
            System.out.println("Bilhetes do Festival Teste 5 : " + festivalTeste5.getNumBilhetes());
            System.out.println();

            System.out.println("####### getNumBilhetes #######");
            System.out.println();
```

```
            System.out.println("####### numActuacoes #######");
            System.out.println();

            System.out.println("Numero de Actuacções de " + artistaTeste1 +
" no Festival Teste 1 > Expectável : 4 | Recebido : " +
festivalTeste1.numActuacoes(artistaTeste1));
            System.out.println("Numero de Actuacções de " + artistaTeste1 +
" no Festival Teste 2 > Expectável : 1 | Recebido : " +
festivalTeste2.numActuacoes(artistaTeste1));
            System.out.println("Numero de Actuacções de " + artistaTeste1 +
" no Festival Teste 3 > Expectável : 0 | Recebido : " +
festivalTeste3.numActuacoes(artistaTeste1));
            System.out.println("Numero de Actuacções de " + artistaTeste1 +
" no Festival Teste 4 > Expectável : 5 | Recebido : " +
festivalTeste4.numActuacoes(artistaTeste1));
            System.out.println("Numero de Actuacções de " + artistaTeste1 +
" no Festival Teste 5 > Expectável : 10 | Recebido : " +
festivalTeste5.numActuacoes(artistaTeste1));
            System.out.println();

            System.out.println("Numero de Actuacções de " + artistaTeste2 +
" no Festival Teste 1 > Expectável : 0 | Recebido : " +
festivalTeste1.numActuacoes(artistaTeste2));
            System.out.println("Numero de Actuacções de " + artistaTeste2 +
" no Festival Teste 2 > Expectável : 0 | Recebido : " +
festivalTeste2.numActuacoes(artistaTeste2));
            System.out.println("Numero de Actuacções de " + artistaTeste2 +
" no Festival Teste 3 > Expectável : 0 | Recebido : " +
festivalTeste3.numActuacoes(artistaTeste2));
            System.out.println("Numero de Actuacções de " + artistaTeste2 +
" no Festival Teste 4 > Expectável : 2 | Recebido : " +
festivalTeste4.numActuacoes(artistaTeste2));
            System.out.println("Numero de Actuacções de " + artistaTeste2 +
" no Festival Teste 5 > Expectável : 2 | Recebido : " +
festivalTeste5.numActuacoes(artistaTeste2));
            System.out.println();

            System.out.println("Numero de Actuacções de " + artistaTeste3 +
" no Festival Teste 1 > Expectável : 0 | Recebido : " +
festivalTeste1.numActuacoes(artistaTeste3));
            System.out.println("Numero de Actuacções de " + artistaTeste3 +
" no Festival Teste 2 > Expectável : 0 | Recebido : " +
festivalTeste2.numActuacoes(artistaTeste3));
            System.out.println("Numero de Actuacções de " + artistaTeste3 +
" no Festival Teste 3 > Expectável : 2 | Recebido : " +
festivalTeste3.numActuacoes(artistaTeste3));
            System.out.println("Numero de Actuacções de " + artistaTeste3 +
" no Festival Teste 4 > Expectável : 2 | Recebido : " +
festivalTeste4.numActuacoes(artistaTeste3));
            System.out.println("Numero de Actuacções de " + artistaTeste3 +
" no Festival Teste 5 > Expectável : 2 | Recebido : " +
festivalTeste5.numActuacoes(artistaTeste3));
            System.out.println();
```

```
            System.out.println("Numero de Actuacções de " + artistaTeste4 +
"  no  Festival  Teste  1  >  Expectável  :  0  |  Recebido  :  " +
festivalTeste1.numActuacoes(artistaTeste4));
            System.out.println("Numero de Actuacções de " + artistaTeste4 +
"  no  Festival  Teste  2  >  Expectável  :  0  |  Recebido  :  " +
festivalTeste2.numActuacoes(artistaTeste4));
            System.out.println("Numero de Actuacções de " + artistaTeste4 +
"  no  Festival  Teste  3  >  Expectável  :  2  |  Recebido  :  " +
festivalTeste3.numActuacoes(artistaTeste4));
            System.out.println("Numero de Actuacções de " + artistaTeste4 +
"  no  Festival  Teste  4  >  Expectável  :  2  |  Recebido  :  " +
festivalTeste4.numActuacoes(artistaTeste4));
            System.out.println("Numero de Actuacções de " + artistaTeste4 +
"  no  Festival  Teste  5  >  Expectável  :  2  |  Recebido  :  " +
festivalTeste5.numActuacoes(artistaTeste4));
            System.out.println();

            System.out.println("Numero de Actuacções de " + artistaTeste5 +
"  no  Festival  Teste  1  >  Expectável  :  0  |  Recebido  :  " +
festivalTeste1.numActuacoes(artistaTeste5));
            System.out.println("Numero de Actuacções de " + artistaTeste5 +
"  no  Festival  Teste  2  >  Expectável  :  0  |  Recebido  :  " +
festivalTeste2.numActuacoes(artistaTeste5));
            System.out.println("Numero de Actuacções de " + artistaTeste5 +
"  no  Festival  Teste  3  >  Expectável  :  2  |  Recebido  :  " +
festivalTeste3.numActuacoes(artistaTeste5));
            System.out.println("Numero de Actuacções de " + artistaTeste5 +
"  no  Festival  Teste  4  >  Expectável  :  2  |  Recebido  :  " +
festivalTeste4.numActuacoes(artistaTeste5));
            System.out.println("Numero de Actuacções de " + artistaTeste5 +
"  no  Festival  Teste  5  >  Expectável  :  2  |  Recebido  :  " +
festivalTeste5.numActuacoes(artistaTeste5));
            System.out.println();

            System.out.println("Numero de Actuacções de " + artistaTeste6 +
"  no  Festival  Teste  1  >  Expectável  :  0  |  Recebido  :  " +
festivalTeste1.numActuacoes(artistaTeste6));
            System.out.println("Numero de Actuacções de " + artistaTeste6 +
"  no  Festival  Teste  2  >  Expectável  :  0  |  Recebido  :  " +
festivalTeste2.numActuacoes(artistaTeste6));
            System.out.println("Numero de Actuacções de " + artistaTeste6 +
"  no  Festival  Teste  3  >  Expectável  :  2  |  Recebido  :  " +
festivalTeste3.numActuacoes(artistaTeste6));
            System.out.println("Numero de Actuacções de " + artistaTeste6 +
"  no  Festival  Teste  4  >  Expectável  :  2  |  Recebido  :  " +
festivalTeste4.numActuacoes(artistaTeste6));
            System.out.println("Numero de Actuacções de " + artistaTeste6 +
"  no  Festival  Teste  5  >  Expectável  :  2  |  Recebido  :  " +
festivalTeste5.numActuacoes(artistaTeste6));
            System.out.println();

            System.out.println("####### numActuacoes #######");
```

```
            System.out.println();

            System.out.println("####### getDeepFestival #######");
            System.out.println();

            System.out.println("DeepFestival do Festival Teste 1 >
Expectável : 0 | Recebido : " + festivalTeste1.getDeepFestival());
            System.out.println("DeepFestival do Festival Teste 2 >
Expectável : 0 | Recebido : " + festivalTeste2.getDeepFestival());
            System.out.println("DeepFestival do Festival Teste 3 >
Expectável : 0 | Recebido : " + festivalTeste3.getDeepFestival());
            System.out.println("DeepFestival do Festival Teste 4 >
Expectável : 1 | Recebido : " + festivalTeste4.getDeepFestival());
            System.out.println("DeepFestival do Festival Teste 5 >
Expectável : 2 | Recebido : " + festivalTeste5.getDeepFestival());
            System.out.println();

            System.out.println("####### getDeepFestival #######");
            System.out.println();

            System.out.println("####### delEvento #######");
            System.out.println();

            String artistaParaRemover1 = "Remover 1";
            String artistaParaRemover2 = "Remover 2";
            String artistaParaRemover3 = "Remover 3";
            String artistaParaRemover4 = "Remover 4";
            String artistaParaRemover5 = "Remover 5";

            Espectaculo espectaculoParaRemover1 = new Espectaculo("Remover
Evento 1", "Testes", 100);
            Espectaculo espectaculoParaRemover2 = new Espectaculo("Remover
Evento 2", "Testes", 300);

            Festival festivalParaRemover1 = new Festival("Remover Evento
1");
            Festival festivalParaRemover2 = new Festival("Remover Evento
2");

            espectaculoParaRemover1.addArtista(artistaParaRemover1);
            espectaculoParaRemover1.addArtista(artistaParaRemover2);
            espectaculoParaRemover2.addArtista(artistaParaRemover3);
            espectaculoParaRemover2.addArtista(artistaParaRemover4);
            espectaculoParaRemover2.addArtista(artistaParaRemover5);

            festivalParaRemover1.addEvento(espectaculoParaRemover2);
            festivalParaRemover2.addEvento(espectaculoParaRemover1);
```

```
            System.out.println("Criação de Eventos para Remover: ");

            System.out.println("Espectáculo 1 : " +
espectaculoParaRemover1.toString());
            System.out.println("Espectáculo 2 : " +
espectaculoParaRemover2.toString());
            System.out.println("Festival 1 : " +
festivalParaRemover1.toString());
            System.out.println("Festival 2 : " +
festivalParaRemover2.toString());
            System.out.println();

            System.out.println("Festival 2 Antes das Adições e Remoções : "
+ festivalTeste2.toString());
            System.out.println();

            System.out.println("Adicionar Festival Remover Evento 1 a
Festival Teste 2 > Expectável : true | Recebido : " +
festivalTeste2.addEvento(festivalParaRemover1));
            System.out.println("Adicionar Festival Remover Evento 2 a
Festival Teste 2 > Expectável : true | Recebido : " +
festivalTeste2.addEvento(festivalParaRemover2));
            System.out.println("Resultado : " + festivalTeste2.toString());
            System.out.println();

            System.out.println("Adicionar Espectaculo Remover Evento 1 a
Festival Teste 2 > Expectável : true | Recebido : " +
festivalTeste2.addEvento(espectaculoParaRemover1));
            System.out.println("Adicionar Espectaculo Remover Evento 2 a
Festival Teste 2 > Expectável : true | Recebido : " +
festivalTeste2.addEvento(espectaculoParaRemover2));
            System.out.println("Resultado : " + festivalTeste2.toString());
            System.out.println();

            System.out.println("Adicionar Festival Remover Evento 2 a
Festival Remover Evento 1 > Expectável : true | Recebido : " +
festivalParaRemover1.addEvento(festivalParaRemover2));
            System.out.println("Resultado : " +
festivalParaRemover1.toString());
            System.out.println("Resultado Festival Teste 2 : " +
festivalTeste2.toString());
            System.out.println();

            System.out.println("Eliminar Festival Remover Evento 1 de
Festival Teste 2 : ");
            System.out.println("Eliminar Festival Remover Evento 1 de
Festival Teste 2 > Expectável : true | Recebido : " +
festivalTeste2.delEvento("Remover Evento 1"));
            System.out.println("Eliminar Festival Remover Evento 1 de
Festival Teste 2 > Expectável : false | Recebido : " +
festivalTeste2.delEvento("Remover Evento 1"));
            System.out.println("Resultado : " + festivalTeste2.toString());
            System.out.println();
```

```
            System.out.println("Eliminar Festival Remover Evento 2 de
Festival Teste 2 > Expectável : true | Recebido : " +
festivalTeste2.delEvento("Remover Evento 2"));
            System.out.println("Eliminar Festival Remover Evento 2 de
Festival Teste 2 > Expectável : false | Recebido : " +
festivalTeste2.delEvento("Remover Evento 2"));
            System.out.println("Resultado : " + festivalTeste2.toString());
            System.out.println();

            System.out.println("####### delEvento #######");
            System.out.println();
      }
}
```